UNIVERSIDADE ESTADUAL PAULISTA
"JÚLIO DE MESQUITA FILHO"
Programa de Pós-Graduação em Ciência da Computação

Luis Henrique Morelli

Arquiteturas Neurais Leves para a Classificação de Boletins Diários de Perfuração em Poços de Petróleo

Bauru - SP
Março de 2026

Luis Henrique Morelli

Arquiteturas Neurais Leves para a Classificação de Boletins Diários de Perfuração em Poços de Petróleo

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru, como parte dos requisitos para a obtenção do título de Mestre, com área de concentração Computação Aplicada e linha de pesquisa Inteligência Computacional.

Fomento: FUNDUNESP - Processo nº 3070/2019

Orientador: Prof. Dr. João Paulo Papa

Bauru - SP
Março de 2026

M842a

Morelli, Luis Henrique
Arquiteturas Neurais Leves para a Classificação de Boletins Diários de Perfuração em Poços de Petróleo / Luis Henrique Morelli. -- Bauru, 2026
106 f. : il., tabs.

Dissertação (mestrado) - Universidade Estadual Paulista (UNESP), Faculdade de Ciências, Bauru
Orientador: João Paulo Papa

1. Processamento de linguagem natural. 2. Inteligência artificial verde. 3. Classificação de textos. 4. Eficiência computacional. 5. Indústria de petróleo e gás. I. Título.

Sistema de geração automática de fichas catalográficas da Unesp. Dados fornecidos pelo autor(a).

UNIVERSIDADE ESTADUAL PAULISTA

Câmpus de Bauru

ATA DA DEFESA PÚBLICA DA DISSERTAÇÃO DE MESTRADO DE LUÍS HENRIQUE MORÉLLI, DISCENTE DO PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO, DA FACULDADE DE CIÊNCIAS - CÂMPUS DE BAURU.

Aos 02 de março de 2026, às 14h30min, por meio de Videoconferência, realizou-se a defesa de DISSERTAÇÃO DE MESTRADO de LUÍS HENRIQUE MORÉLLI, intitulada "Arquiteturas Neurais Leves para a Classificação de Boletins Diários de Perfuração em Poços de Petróleo". A Comissão Examinadora foi constituída pelos seguintes membros: Prof. Dr. JOÃO PAULO PAPA (Orientador - Participação Virtual) do Departamento de Computação /UNESP/Câmpus de Bauru - FC, Professor Doutor KELTON AUGUSTO PONTARA DA COSTA (Participação Virtual) do Departamento de Computação/UNESP/Câmpus de Bauru - FC, Prof. Dr. JOÃO BAPTISTA CARDIA NETO (Participação Virtual) da Faculdade de Tecnologia de Catanduva/Centro Estadual de Educação Tecnológica Paula Souza. Após a exposição pelo mestrando e arguição pelos membros da Comissão Examinadora que participaram do ato, de forma virtual, o discente recebeu o conceito final: APROVADO. Nada mais havendo, foi lavrada a presente ata, que após lida e aprovada, foi assinada pelo Presidente da Comissão Examinadora.

gov.br
Documento assinado digitalmente
JOAO PAULO PAPA
Data: 02/03/2026 17:02:46-0300
Verifique em https://validar.iti.gov.br

Prof. Dr. JOÃO PAULO PAPA

Luis Henrique Morelli

Arquiteturas Neurais Leves para a Classificação de Boletins Diários de Perfuração em Poços de Petróleo

Dissertação apresentada ao Programa de Pós-Graduação em Ciência da Computação da Faculdade de Ciências da Universidade Estadual Paulista "Júlio de Mesquita Filho", Câmpus de Bauru, como parte dos requisitos para a obtenção do título de Mestre, com área de concentração Computação Aplicada e linha de pesquisa Inteligência Computacional.

Fomento: FUNDUNESP - Processo 3070/2019

Banca Examinadora

Prof. Dr. João Paulo Papa
Universidade Estadual Paulista "Júlio de Mesquita Filho" - Câmpus de Bauru
Orientador

Prof. Dr. Kelton Augusto Pontara da Costa
Universidade Estadual Paulista "Júlio de Mesquita Filho" - Câmpus de Bauru

João Baptista Cardia Neto
Centro Estadual de Educação Tecnológica "Paula Souza" - Faculdade de Tecnologia de Catanduva

Bauru - SP
02 de março de 2026

Dedico este trabalho à minha família, ao meu cachorrinho Dante, aos meus professores e amigos, bem como a todos que se fizeram presentes ao longo desta jornada.

Agradecimentos

Agradeço, primeiramente, àquele ou àquilo que denominamos Deus, força criadora do universo e das leis que o regem, por conceder-me o dom da vida e a oportunidade de vivenciar todas as experiências e emoções dela decorrentes.

Agradeço imensamente à minha família, aos meus pais, Luis Fabiano Morelli e Marta Mônica da Silva Morelli, e à minha irmã, Monique Vitória Morelli, por todo amor, carinho, apoio, sustento, lições, motivação, e por tudo aquilo que as palavras não conseguem plenamente expressar. Vocês são minha base, meu alicerce e meu tudo; um simples agradecimento, ou mesmo a imensidão do cosmos, não são suficientes para dimensionar o que sinto por vocês.

Agradeço à minha avó, Helena Piovesana Morelli Hihihi, por sempre me receber com o mais belo sorriso e por nunca deixar de comentar o quanto eu estava bonito nos dias em que a visitava; por fazer questão de dizer, com orgulho, a todas as pessoas, que eu era seu neto; por colocar um sorriso em meu rosto apenas com a palavra "É...", sem falar nas gargalhadas provocadas pelas piadinhas; por me ensinar, de forma inesquecível, o que realmente acontece quando o requeijão estraga; por me fazer ver o bem nas pessoas; e, sobretudo, por tudo o que fez por mim e pela minha educação ao longo de toda a minha vida. Que você esteja brilhando intensamente e cuidando dos jardins mais floridos e bonitos no além-vida.

Agradeço a todos os meus professores do ensino fundamental, do ensino médio, da graduação e do mestrado, por pavimentarem minha trajetória no caminho do conhecimento. Em especial, agradeço ao Prof. Dr. João Paulo Papa, pela oportunidade de trabalharmos juntos neste mestrado, bem como por toda a orientação, confiança e aprendizado compartilhados ao longo deste período, e ao Prof. Dr. Ivan Rizzo Guilherme, pela oportunidade de colaborar em projetos desafiadores e pela perspectiva transformadora que me proporcionou. Suas provocações intelectuais foram essenciais para o amadurecimento do meu pensamento, permitindo-me encarar tanto a ciência quanto a vida sob novos prismas.

Agradeço a toda a equipe do projeto "ProtoRADIAR: Métodos de Captura e Disseminação do Conhecimento, por meio de Processamento de Linguagem Natural na Área de Poços", pelo acolhimento e pela parceria. Sou grato pela confiança em meu trabalho e, sobretudo, pelos debates que confrontaram minhas ideias e me indicaram caminhos mais assertivos, em um ambiente de autonomia e colaboração que sempre busquei. Ademais, agradeço à Fundunesp pela concessão da bolsa de estudos, viabilizando a dedicação para a realização deste trabalho.

Por fim, agradeço aos meus amigos e colegas de curso que tornaram este percurso mais leve, alegre e memorável. Em especial, agradeço a Davi Augusto Neves Leite, Giovani Candido e Luiz Fernando Merli de Oliveira Sementille, que me acompanham desde a graduação. Obrigado por dividirem comigo os desafios e as glórias da vida acadêmica e pela amizade resiliente.

"Knowing your own ignorance is the first step to enlightenment."
Patrick Rothfuss

Resumo

A classificação automatizada de Boletins Diários de Perfuração na indústria de petróleo e gás, embora beneficiada pelos avanços em modelos de Deep Learning, enfrenta barreiras críticas associadas ao elevado custo computacional, alta latência e consumo energético proibitivo dos modelos tradicionais baseados em Transformers. Diante desse cenário, esta dissertação teve como objetivo principal desenvolver e validar arquiteturas neurais de complexidade reduzida, investigando a hipótese de que modelos leves, aliados a técnicas de otimização, podem manter desempenho competitivo com maior viabilidade operacional. A abordagem metodológica consistiu em uma análise comparativa sistemática de arquiteturas eficientes, incluindo CNN, BiLSTM e DistilBERT, potencializadas por técnicas como Destilação de Conhecimento e Mixture of Experts, sob um protocolo experimental que integrou métricas de eficácia preditiva e eficiência energética. Os resultados demonstraram que as abordagens propostas reduziram drasticamente o tempo de inferência e a pegada de carbono, preservando a acurácia e o F1-Score em níveis estatisticamente equivalentes ou superiores aos baselines densos. Conclui-se que a adoção de estratégias de Green AI viabiliza a implementação de soluções robustas, sustentáveis e escaláveis para a automação de processos decisórios em ambientes industriais com restrições de hardware, confirmando a eficácia da compressão de modelos para o domínio de óleo e gás.

Palavras-chave: Processamento de Linguagem Natural, Inteligência Artificial Verde, Classificação de Textos, Eficiência Computacional, Indústria de Petróleo e Gás.

Abstract

The automated classification of Daily Drilling Reports in the oil and gas industry, while benefiting from advances in Deep Learning models, faces critical barriers associated with high computational costs, latency, and prohibitive energy consumption of traditional Transformer-based models. In this context, this dissertation aimed to develop and validate reduced-complexity neural architectures, investigating the hypothesis that lightweight models, combined with optimization techniques, can maintain competitive performance with greater operational viability. The methodological approach consisted of a systematic comparative analysis of efficient architectures, including CNN, BiLSTM, and DistilBERT, enhanced by techniques such as Knowledge Distillation and Mixture of Experts, under an experimental protocol integrating predictive efficacy and energy efficiency metrics. The results demonstrated that the proposed approaches drastically reduced inference time and carbon footprint while preserving accuracy and F1-Score at levels statistically equivalent or superior to dense baselines. It is concluded that adopting Green AI strategies enables the implementation of robust, sustainable, and scalable solutions for automating decision-making processes in hardware-constrained industrial environments, confirming the efficacy of model compression for the oil and gas domain.

Keywords: Natural Language Processing, Green AI, Text Classification, Computational Efficiency, Oil and Gas Industry.

Lista de Figuras

Lista de Tabelas

Lista de Abreviaturas e Siglas

ANN	Artificial Neural Network
API	Application Program Interface
BDP	Boletim Diário de Perfuração
BERT	Bidirectional Encoder Representations from Transformers
BiLSTM	Bidirectional Long Short-Term Memory
CE	Cross-Entropy Loss
CENPES	Centro de Pesquisas Leopoldo Américo Miguez de Mello
CNN	Convolutional Neural Network
CPU	Central Processing Unit
CRF	Conditional Random Fields
CV	Coeficiente de Variação
DAPT	Domain Adaptive Pre-training
DL	Deep Learning
DNN	Deep Neural Network
FLOP	*Floating Point Operations*
FLOPS	*Floating Point Operations per Second*
FN	Falso Negativo
FNN	Feedforward Neural Network
FP	Falso Positivo
FUNDUNESP	Fundação para o Desenvolvimento da UNESP
FWER	Family-Wise Error Rate
GELU	Gaussian Error Linear Unit
GNN	Graph Neural Network

GPU	Graphics Processing Unit
GRU	Bidirectional Gated Recurrent Unit
HPO	Hyperparameter Optimization
IA	Inteligência Artificial
IoT	Internet of Things
KD	Knowledge Distillation
LSTM	Long Short-Term Memory
ML	Machine Learning
MLP	Multi-Layer Perceptron
MoE	Mixture of Experts
MSE	Mean-Squared Error
NCE	Noise-Contrastive Estimation
NER	Named Entity Recognition
NLU	Natural Language Understanding
NVML	NVIDIA Management Library
OOD	Out-Of-Distribution
Petrobras	Petróleo Brasileiro S.A.
PG	Programação Genética
PLM	Pre-trained Language Model
PLN	Processamento de Linguagem Natural
Patient-KD	Patient Knowledge Distillation
QAT	Quantization Aware Training
QLoRA	Quantized Low-Rank Adaptation
RAPL	Running Average Power Limit
RBLP	Rule-Based Language Processing
ReLU	Rectified Linear Unit

RL	Reinforcement Learning
RNN	Recurrent Neural Network
SIGITEC	Sistema de Gestão de Investimentos em Tecnologia
SNN	Shallow Neural Network
TAPT	Task Adaptive Pre-Training
TPU	Tensor Processing Unit
UNESP	Universidade Estadual Paulista "Júlio de Mesquita Filho"
VP	Verdadeiro Positivo

Sumário

1 Introdução

As operações de perfuração de poços em ambientes terrestres e marítimos consistem, fundamentalmente, na fragmentação de formações rochosas para a abertura e o aprofundamento de poços destinados à extração de petróleo e gás natural. Conforme enfatizado por Ribeiro et al. (2020), tais atividades caracterizam-se por serem demoradas, custosas e complexas, representando uma etapa crítica no desenvolvimento de campos petrolíferos. Essa importância estratégica, como apontam Ma et al. (2018), desperta o interesse constante de empresas de energia, órgãos reguladores e da sociedade civil. Nesse contexto, a fim de cumprir exigências normativas, mitigar riscos ambientais e de segurança do trabalho e otimizar os processos, as organizações do setor implementam procedimentos padronizados. Essas normas estabelecem a obrigatoriedade do registro sistemático das atividades, visando garantir a manutenção de um histórico detalhado e fidedigno de todas as intervenções realizadas.

Por conseguinte, o Boletim Diário de Perfuração (BDP) destaca-se como o principal instrumento para o registro sistemático de informações operacionais e eventos observados durante as atividades de perfuração. Esses documentos consolidam análises elaboradas por perfuradores a partir de medições obtidas por sensores de fundo e equipamentos de superfície. No entanto, a análise desses textos em larga escala apresenta desafios significativos, conforme discutido por Hoffimann et al. (2018). Complementando essa visão, Ma et al. (2018) ressaltam que a complexidade da tarefa decorre da vasta extensão dos conjuntos de dados, da ausência de estruturação, da ocorrência frequente de lacunas e erros, além da subjetividade e informalidade do vocabulário utilizado pelos relatores. Tais características tornam a classificação manual e precisa das atividades um obstáculo considerável para equipes humanas com recursos limitados. Diante dessa realidade, Hoffimann et al. (2018) sugerem que essa tarefa seja idealmente atribuída a sistemas inteligentes de relatórios capazes de operar em tempo real. Tal automatização permitiria que profissionais de campo, engenheiros e gestores interpretassem tendências prontamente, identificassem padrões de falha e adotassem medidas corretivas ágeis, otimizando o diagnóstico de problemas e o monitoramento das operações.

Nos últimos anos, o interesse por técnicas de aprendizado de máquina, ou Machine Learning (ML) em inglês, e aprendizado profundo, ou Deep Learning (DL) em inglês, expandiu-se consideravelmente, fundamentado em progressos notáveis em campos como visão computacional e Processamento de Linguagem Natural (PLN). Essas abordagens têm sido aplicadas na resolução de diversos problemas reais que envolvem o processamento de texto, imagem, áudio e vídeo, abrangendo tarefas de tradução, detecção de objetos, reconhecimento de fala e classificação de dados. Nesse cenário, Sousa et al. (2018) destacam a relevância da classificação automatizada de documentos, impulsionada pela abundância de textos gerados e consumidos cotidianamente. Corroborando essa visão, Ribeiro et al. (2020) observam que os relatórios de

perfuração constituem uma fonte rica de informações, com elevado potencial para a análise de padrões e a mineração de texto no domínio de petróleo e gás natural por meio de DL, conforme evidenciado também em pesquisas recentes (HOFFIMANN et al., 2018; RIBEIRO et al., 2020; CINELLI et al., 2021; RODRIGUES et al., 2022).

As metodologias propostas por Ribeiro et al. (2020) e Rodrigues et al. (2022) fundamentam frentes de pesquisa integradas ao projeto "ProtoRADIAR: Métodos de Captura e Disseminação do Conhecimento por meio de Processamento de Linguagem Natural na Área de Poços"[1]. Essa iniciativa é promovida pelo Centro de Pesquisas Leopoldo Américo Miguez de Mello (CENPES), unidade de pesquisa e desenvolvimento da Petróleo Brasileiro S.A. (Petrobras), em colaboração com a Universidade Estadual Paulista 'Júlio de Mesquita Filho" (UNESP), por intermédio do laboratório Recogna[2], sediado no *campus* de Bauru. Inserida nesse contexto, a presente pesquisa é desenvolvida pelo autor como integrante do referido laboratório e vinculada ao projeto ProtoRADIAR. Um dos desafios prementes enfrentados atualmente reside na implementação de modelos de DL em tarefas que exigem processamento em tempo real, como a classificação de relatórios elaborados por assistentes de preenchimento de boletins, dada a elevada carga computacional demandada pelas redes neurais profundas, ou Deep Neural Network (DNN) em inglês.

Nesse cenário, observa-se que a evolução do estado da arte em DL está intrinsecamente vinculada ao aprimoramento em *benchmarks*, padrões estabelecidos para mensurar objetivamente a qualidade da aprendizagem em tarefas como classificação de imagens e textos, o que tem resultado no aumento progressivo da complexidade das redes neurais. Tal complexidade manifesta-se na quantidade de parâmetros dos modelos e na consequente demanda por recursos computacionais de alto desempenho, como as unidades de processamento gráfico, ou Graphics Processing Unit (GPU) em inglês, impactando diretamente a latência durante as inferências. Portanto, embora esses modelos apresentem desempenho satisfatório sob condições controladas, eles frequentemente carecem da eficiência necessária para a implantação direta em cenários reais, o que impõe desafios significativos ao desenvolvimento de aplicações práticas. Muitas demandas do mundo real exigem tomadas de decisão em tempo real, de forma análoga à capacidade humana; na condução autônoma, por exemplo, o sistema deve ser capaz de detectar localmente pedestres, animais e obstáculos, além de calcular distâncias instantaneamente, visando garantir a segurança da operação sem intervenção humana.

Dada a crescente complexidade das arquiteturas de DL, a redução do número de parâmetros e a concepção de modelos compactos aptos a operar em dispositivos com restrições de *hardware* tornaram-se frentes de investigação fundamentais (CHOUDHARY et al., 2020). Redes neurais superparamatrizadas resultam em maior latência de inferência e elevado consumo energético e de memória, o que limita sua portabilidade em comparação a modelos otimizados.

[1] Processo 2019/00697-8 - Sistema de Gestão de Investimentos em Tecnologia (SIGITEC)
[2] Disponível em: https://recogna.tech. Acesso em: 01 ago. 2024

Segundo Cheng et al. (2018), a viabilização dessas soluções eficientes exige uma abordagem multidisciplinar que integre avanços em ML, otimização, compressão de dados e *design* de *hardware*. Nesse panorama, diversas técnicas para a otimização de DNNs têm sido propostas, abrangendo métodos de compressão como poda, ou *pruning* em inglês, e quantização, além da destilação de conhecimento, ou Knowledge Distillation (KD) em inglês, da Busca por Arquitetura Neural, ou *Neural Architecture Search* (NAS) em inglês, e do desenvolvimento de arquiteturas inerentemente eficientes, a exemplo da Mixture of Experts (MoE).

Sendo assim, este estudo propõe uma alternativa eficiente para a classificação de BDPs, fundamentada no treinamento de diferentes arquiteturas de DL. A problemática é abordada sob a perspectiva conjunta da eficácia na classificação e da eficiência computacional, buscando equilibrar o desempenho preditivo e o consumo de recursos. Para a validação da abordagem proposta, realiza-se a comparação entre a qualidade das predições de risco obtidas e os valores reais de deterioração, permitindo mensurar a fidedignidade das estimativas em relação aos dados observados.

1.1 Problema de pesquisa

A crescente complexidade e o contínuo aumento no volume de dados gerados na indústria de petróleo e gás, notadamente os BDPs, impõem desafios significativos à análise e à interpretação eficazes desses documentos. A classificação automatizada de tais registros é fundamental para embasar a tomada de decisão e otimizar os processos operacionais. Contudo, a adoção de modelos de DL frequentemente esbarra na alta demanda por recursos de *hardware*, o que contrasta com a necessidade premente de soluções computacionalmente eficientes. Nesse contexto, esta pesquisa visa contribuir com os trabalhos do grupo ProtoRADIAR ao investigar a aplicação de arquiteturas neurais leves na classificação de BDPs, buscando conciliar a precisão na categorização com a eficiência necessária para a operação em ambientes de recursos restritos. Por conseguinte, o estudo propõe-se a responder às seguintes questões de pesquisa:

- As arquiteturas propostas alcançam desempenho competitivo frente aos modelos anteriormente empregados na resolução deste problema?
- O incremento na eficiência computacional justifica o emprego das abordagens propostas em infraestruturas com recursos de processamento limitados, mesmo diante de eventuais perdas na capacidade preditiva?

1.2 Hipótese

A hipótese central deste estudo estabelece que a implementação de arquiteturas neurais, integrada a técnicas de compressão e otimização de modelos, proporcionará uma classificação

de BDPs mais eficiente e precisa em comparação aos modelos de DL aplicados atualmente a esse problema no âmbito do projeto. De modo específico, pressupõe-se que:

1. As arquiteturas neurais leves, em virtude de sua estrutura simplificada e do reduzido volume de parâmetros, devem proporcionar uma redução expressiva no custo computacional e na latência de inferência, tornando-as adequadas para ambientes operacionais com restrições de *hardware*. Tal otimização permitirá que a classificação dos BDPs ocorra em tempo real, facilitando a tomada de decisões ágeis e fundamentadas no contexto das atividades de campo;

2. Espera-se que a redução na complexidade arquitetural não comprometa significativamente o desempenho, avaliado em termos de precisão e revocação, na classificação dos BDPs. A aplicação de técnicas de compressão, a exemplo da KD, em conjunto com arquiteturas eficientes como a MoE, tem o objetivo de preservar ao máximo a capacidade de generalização do modelo. Tal estratégia busca minimizar eventuais perdas de eficácia quando comparada à utilização de modelos densos e robustos, de modo a assegurar que a otimização de recursos computacionais ocorra simultaneamente à manutenção da qualidade preditiva exigida pela operação;

3. A adoção de classificadores eficientes não apenas atende às demandas operacionais da indústria de petróleo e gás, mas também contribui para a sustentabilidade e a inovação na gestão de informações. Tal abordagem permite uma integração mais fluida com os sistemas existentes, além de facilitar os processos de atualização e manutenção dos modelos, garantindo maior agilidade e perenidade às soluções desenvolvidas.

1.3 Objetivo geral

Nessa conjuntura, o objetivo principal deste estudo consiste em desenvolver e validar arquiteturas neurais de complexidade reduzida para a classificação automatizada de BDPs, visando mitigar o custo computacional e a latência de processamento inerentes aos modelos de DL convencionais. A pesquisa pretende não apenas atender às necessidades específicas da indústria de petróleo e gás, mas também consolidar um arcabouço teórico e prático aplicável a outros domínios nos quais a conciliação entre eficiência e precisão na classificação de dados seja um requisito crítico.

1.4 Objetivos específicos

No que tange às metas específicas, este projeto almeja:

- Selecionar e otimizar arquiteturas de redes neurais eficientes, empregando técnicas de compressão e otimização, visando assegurar que a busca pela eficiência computacional não comprometa o desempenho necessário para a classificação precisa dos BDPs.
- Realizar uma análise comparativa entre as arquiteturas neurais leves e os modelos atualmente utilizados na indústria, por meio de métricas de desempenho que integrem a precisão classificatória e a eficiência computacional. Tal análise deve contemplar indicadores específicos, como o tempo de inferência e o consumo de recursos de *hardware*, com o objetivo de mensurar objetivamente o equilíbrio entre o poder preditivo e a viabilidade técnica das soluções em ambientes de produção;
- Contribuir para a prática industrial mediante a proposição de um modelo de classificação de fácil integração aos sistemas vigentes na indústria de petróleo e gás, fomentando a automação e a eficiência na análise de dados operacionais. Essa iniciativa visa otimizar o processamento de informações e reduzir a latência na interpretação de registros de campo, consolidando a aplicabilidade prática das arquiteturas neurais de complexidade reduzida desenvolvidas neste estudo;
- Promover a disseminação das contribuições alcançadas por meio do compartilhamento dos resultados da pesquisa em eventos acadêmicos e publicações científicas, colaborando para o avanço do conhecimento no campo de DL e para o fortalecimento de suas aplicações práticas em contextos industriais.

1.5 Organização da monografia

A estrutura desta dissertação está organizada em cinco capítulos principais, cujos conteúdos são brevemente descritos a seguir:

Capítulo 2: Apresenta uma revisão abrangente da literatura, fundamentando o problema de pesquisa por meio da análise de metodologias e estudos anteriores, com ênfase na identificação das lacunas que justificam a investigação proposta.

Capítulo 3: Detalha os procedimentos experimentais, incluindo a descrição das bases de dados, a abordagem metodológica adotada e o projeto das arquiteturas neurais leves, além de especificar as técnicas de compressão e otimização aplicadas.

Capítulo 4: Apresenta e discute os resultados obtidos, estabelecendo uma análise comparativa entre as arquiteturas propostas e os modelos de referência, considerando métricas de desempenho preditivo e eficiência computacional em cenários industriais.

Capítulo 5: Sintetiza as principais contribuições da pesquisa, verifica o cumprimento dos objetivos estabelecidos e propõe direcionamentos para trabalhos futuros voltados à otimização de modelos em ambientes de produção.

2 Revisão de Literatura

Este capítulo fundamenta teórica e tecnicamente a investigação, estabelecendo o estado da arte para a classificação de textos em domínios especializados e as estratégias para a sua implementação eficiente. A discussão inicia-se pela análise dos desafios inerentes ao PLN em contextos de vocabulário técnico e restrito, como a indústria de petróleo e gás, o setor financeiro e a esfera governamental. Embora o foco da aplicação recaia sobre os BDPs, examinam-se abordagens transversais que lidam com a escassez de dados anotados e a complexidade terminológica.

Em resposta à crescente complexidade dos modelos de DL necessários para estas tarefas, aborda-se subsequentemente o paradigma de Green AI. Esta secção discute a importância crítica da eficiência computacional e da sustentabilidade, contrapondo a tendência de modelos massivos às restrições impostas por ambientes industriais e dispositivos de hardware limitado. Por fim, o capítulo detalha as metodologias selecionadas para conciliar alto desempenho preditivo com baixo custo computacional. São examinados os fundamentos da KD, que permite a transferência de competências de modelos robustos para arquiteturas leves, e das arquiteturas MoE, que introduzem a ativação condicional e esparsa de parâmetros. A compreensão integrada destes tópicos é essencial para contextualizar a proposta de modelos compactos capazes de operar em tempo real sem comprometer a precisão da classificação.

2.1 Classificação de textos em domínios especializados

Inspirados por um estudo seminal que aplicava técnicas convencionais de PLN em BDPs, Hoffimann et al. (2018) abordaram os desafios analíticos impostos pelo vasto volume desses relatórios na indústria de petróleo e gás. Embora fundamentais para a mitigação de acidentes e a otimização operacional, tais registros apresentam uma estrutura textual complexa, caracterizada por terminologias técnicas, abreviações e ruídos que dificultam a extração de informações em larga escala. Para mitigar essa problemática, os autores propuseram uma metodologia fundamentada em PLN profundo, integrando o pré-processamento via expressões regulares ao uso de *word embeddings* gerados pelo modelo skip-gram (MIKOLOV et al., 2011) com Noise-Contrastive Estimation (NCE) (GUTMANN; HYVäRINEN, 2010) para capturar a semântica técnica do domínio.

A abordagem avaliou três arquiteturas de redes neurais para a classificação de sentenças em categorias de EVENTO, SINTOMA e AÇÃO: (i) Multi-Layer Perceptron (MLP) simples com média aritmética; (ii) redes neurais convolucionais, ou Convolutional Neural Networks (CNN) em inglês (LECUN; BENGIO, 1998); e (iii) redes de memória de longo e curto prazo, ou Long Short-Term Memory (LSTM) em inglês (HOCHREITER; SCHMIDHUBER, 1997). Os resultados

experimentais evidenciaram a superioridade do modelo LSTM, que alcançou uma acurácia média de 82,7% em validação cruzada de 5 *folds*, superando as demais arquiteturas por sua capacidade de modelar o contexto sequencial do texto, em consonância com a literatura da área. A principal contribuição da pesquisa reside na demonstração de que o aprendizado profundo pode automatizar a análise de sequências operacionais e identificar padrões de falhas subutilizados pela indústria. Contudo, o estudo aponta limitações relacionadas ao desbalanceamento dos dados e à ocorrência de classificações incorretas entre eventos e sintomas, ressaltando que o refinamento da rotulagem por especialistas e a expansão do *corpus* técnico são premissas fundamentais para o aprimoramento da robustez do sistema em cenários de tempo real.

Em uma perspectiva complementar, Ribeiro et al. (2020) investigaram a classificação automática de eventos em BDPs, enfrentando desafios inerentes à natureza não estruturada dos textos, à terminologia técnica específica e à forte dependência sequencial entre as operações de perfuração. A metodologia proposta baseia-se em uma arquitetura de redes neurais recorrentes, ou Recurrent Neural Network (RNN) em inglês, que utiliza unidades recorrentes fechadas bidirecionais, ou Bidirectional Gated Recurrent Unit (BiGRU) em inglês, integradas a mecanismos de atenção para a extração de características semânticas, combinadas a uma camada de Conditional Random Fields (CRF) (LAFFERTY; MCCALLUM; PEREIRA, 2001) para modelar as interdependências entre rótulos adjacentes na sequência operacional. Uma contribuição central do trabalho reside na proposição de um *ensemble* ponderado por algoritmos evolutivos, especificamente Algoritmos Genéticos (GOLDBERG; HOLLAND, 1988) e Programação Genética (KOZA, 1992), cujos membros são gerados a partir de estados de ótimos locais obtidos via taxas de aprendizado cíclicas.

Resultados experimentais em bases de dados reais e desbalanceadas da Petrobras demonstraram que o modelo contextual superou o *baseline fastText* (JOULIN et al., 2017) em 47% na métrica Macro-F1, com o *ensemble* evolutivo proporcionando incrementos adicionais de até 3%. Contudo, os autores observaram que o desempenho do sistema é sensível à extensão das sequências e à frequência de transições entre operações, ressaltando que blocos de sentenças reduzidos podem limitar a capacidade de aprendizado e que a segmentação prévia dos relatórios é necessária para viabilizar o processamento de dependências de longo prazo.

Sob uma perspectiva distinta, voltada à superação das limitações das arquiteturas recorrentes anteriormente discutidas, Cinelli et al. (2021) e Rodrigues et al. (2022) exploraram o emprego de Transformers para a extração e identificação automatizada de eventos em BDPs. No estudo de Cinelli et al. (2021), a metodologia fundamentou-se na comparação entre um sistema especialista baseado em regras, ou Rule-Based Language Processing (RBLP) em inglês, que utiliza protocolos de busca sequencial com expressões regulares, e uma abordagem de DL baseada no modelo de representações de codificador bidirecional dos Transformers, ou Bidirectional Encoder Representations from Transformers (BERT) (DEVLIN et al., 2019).

A pesquisa inova ao caracterizar a análise de BDPs como um problema de classificação

multi-rótulo, permitindo a identificação simultânea de múltiplas falhas operacionais, como *stuck pipe*, *kick* e perda de circulação, o que representa um avanço frente a modelos restritos a causas únicas de tempo não produtivo. Experimentalmente, os autores observaram que o método RBLP atingiu uma taxa média de verdadeiros positivos de 97,30%, enquanto o modelo BERT obteve 85,61%. No entanto, ao desconsiderar classes sub-representadas e raras, o desempenho da rede neural elevou-se para 97,32%, superando levemente a técnica baseada em regras. Concluiu-se que, embora o RBLP demonstre maior robustez em cenários de escassez de dados e desbalanceamento de classes, a abordagem fundamentada em Transformers proporciona maior flexibilidade operacional por meio do ajuste de limiares de probabilidade, apesar de sua maior complexidade computacional e dependência de grandes volumes de dados anotados.

Dando prosseguimento ao uso de modelos baseados em Transformers, Rodrigues et al. (2022) propuseram o PetroBERT, um modelo de linguagem especializado no domínio de exploração de petróleo e gás em língua portuguesa. O objetivo central do trabalho foi mitigar as limitações de modelos generalistas ao lidar com a ambiguidade linguística e o vocabulário técnico específico do setor. A abordagem metodológica fundamenta-se na técnica de pré-treinamento adaptativo de domínio, ou Domain Adaptive Pre-Training (DAPT) em inglês, das arquiteturas BERT Multilíngue (mBERT) (DEVLIN et al., 2019) e BERTimbau (SOUZA; NOGUEIRA; LOTUFO, 2020a), utilizando, para esse fim, o *corpus* Petrolês e um conjunto privado de relatórios diários de perfuração da Petrobras.

A pesquisa destaca-se pelo pioneirismo ao introduzir o primeiro modelo BERT voltado ao contexto petrolífero, avaliado em tarefas de Reconhecimento de Entidades Nomeadas, ou Named Entity Recognition (NER) em inglês, e classificação de sentenças em múltiplos níveis de detalhamento operacional. Os resultados experimentais demonstraram que a adaptação foi particularmente eficaz na classificação de sentenças, tarefa na qual a variante PetroBERT$_{pt_ddr}$ alcançou um F1-score de 51,98%, corroborando a premissa de que modelos treinados especificamente em português superam versões multilíngues em tarefas do mesmo idioma. Contudo, os autores apontam que restrições temporais e de infraestrutura computacional limitaram o treinamento a poucas épocas em uma única GPU, sugerindo que o desempenho do modelo pode ser potencializado em cenários de processamento de maior escala.

Expandindo a investigação sobre a eficácia de modelos especializados para domínios específicos, Silva et al. (2024) exploraram os desafios da aplicação de modelos de linguagem natural no contexto governamental brasileiro. Nesse cenário, o vocabulário técnico e as estruturas sintáticas burocráticas frequentemente resultam em um desempenho subótimo de modelos treinados em *corpora* genéricos. Para endereçar essa lacuna, os autores empregaram a técnica de DAPT, desenvolvendo e avaliando sistematicamente 18 modelos baseados nas arquiteturas BERTimbau e LaBSE. A metodologia envolveu a variação controlada de três fatores críticos: a relevância do conjunto de dados de destino, abrangendo segmentos de diários oficiais, documentos jurídicos e itens de despesas eleitorais; a composição linguística, dividida entre as

vertentes monolíngue e multilíngue; e a escala dos dados de pré-treinamento.

Os resultados, mensurados pela métrica Macro-F1 em tarefas de classificação, demonstram que a eficácia da adaptação está fortemente vinculada à similaridade linguística entre o *corpus* de pré-treinamento e a tarefa final, com os modelos derivados do BERTimbau apresentando, em média, superioridade em relação ao LaBSE em contextos do setor público. Além disso, a pesquisa revelou que o incremento quantitativo de dados durante o DAPT não garante melhorias lineares no desempenho em tarefas de ajuste fino, ressaltando que a qualidade e a representatividade dos dados prevalecem sobre o volume bruto. Embora o trabalho contribua para o estado da arte em PLN aplicado ao governo, os autores reconhecem limitações decorrentes da ausência de testes de significância estatística, devido a restrições computacionais, e do persistente desbalanceamento de classes nos conjuntos de dados analisados.

De forma análoga às investigações no setor público, Peng et al. (2021) avaliaram a eficácia da adaptação de domínio para arquiteturas baseadas em Transformers no setor financeiro, buscando determinar se o pré-treinamento especializado supera modelos generalistas em tarefas que transcendem a análise de sentimento convencional. A metodologia compreendeu uma análise comparativa entre o modelo BERT original e duas variantes do FinBERT: uma fundamentada no pré-treinamento continuado com a preservação do vocabulário geral, e outra treinada integralmente do zero com a incorporação de um vocabulário financeiro específico.

Ao expandir o escopo experimental para incluir a detecção de causalidade e o processamento de numerais, os autores demonstraram que o pré-treinamento continuado apresentou resultados mais consistentes, superando o BERT na maioria dos *benchmarks* e destacando-se na compreensão de termos numéricos. A principal contribuição do trabalho reside na evidência de que a exposição a textos do domínio é mais determinante para o sucesso do modelo do que a customização do vocabulário, divergindo de resultados observados na área biomédica. Contudo, o estudo aponta limitações como a instabilidade do ajuste fino em bases de dados reduzidas e a dificuldade em interpretar nuances semânticas complexas, a exemplo da ironia e de construções gramaticais de finalidade que mimetizam relações de causalidade.

2.2 Eficiência computacional e Green AI

Nas últimas décadas, o DL fundamentado em DNNs consolidou-se como a abordagem predominante para o desenvolvimento de modelos de aprendizado de máquina. Conforme apontam Cheng et al. (2018), Choudhary et al. (2020) e Menghani (2021), o DL ganhou projeção global impulsionado pela competição ImageNet (RUSSAKOVSKY et al., 2015) de 2012, na qual a AlexNet (KRIZHEVSKY; SUTSKEVER; HINTON, 2012), uma CNN profunda, alcançou resultados inovadores. Essa arquitetura era composta por 60 milhões de parâmetros, estruturados em cinco camadas convolucionais e três camadas totalmente conectadas. Nos anos seguintes, modelos como VGGNet (SIMONYAN; ZISSERMAN, 2015), Inception (SZEGEDY

et al., 2015) e ResNet (HE et al., 2016) superaram sucessivamente os recordes de desempenho no desafio ImageNet.

Esse impacto estendeu-se ao campo da compreensão de linguagem natural, ou Natural Language Understanding (NLU) em inglês, no qual a arquitetura Transformers fomentou a criação de codificadores de propósito geral, a exemplo do BERT e do GPT-3 (BROWN et al., 2020). Enquanto o BERT estabeleceu novos patamares de desempenho em diversos *benchmarks* de linguagem, o GPT-3 consolidou-se na indústria por meio de sua interface de programação de aplicações, ou Application Programming Interface (API) em inglês. O denominador comum entre esses avanços reside no crescimento acelerado da escala dos modelos e no consequente aumento nos custos de treinamento e implantação.

Em decorrência dos avanços mencionados, consolidou-se uma tendência de desenvolvimento de DNNs progressivamente mais complexas e com maior volume de parâmetros. Contudo, Deng et al. (2020) e Mishra, Gupta e Dutta (2020) advertem que a elevada precisão das DNNs é obtida sob um alto custo de memória e complexidade computacional, exigindo recursos consideráveis de energia, processamento e armazenamento. Adicionalmente, Deng et al. (2020) argumentam que a rápida expansão dessas arquiteturas foi viabilizada pelo aprimoramento do desempenho de processadores modernos, em especial das GPUs em inglês.

Na visão dos autores, a ascensão das GPUs como plataforma central deve-se ao fato de que as unidades centrais de processamento, ou Central Processing Units (CPUs) em inglês, mostram-se incapazes de suprir as crescentes demandas por largura de banda de memória e poder de cálculo impostas pelo aumento contínuo dos modelos de DL. Atualmente, uma DNN pode ultrapassar 10.000 camadas e conter milhões ou bilhões de parâmetros e estados intermediários (DENG et al., 2020). Diante do elevado custo operacional dessas redes, as GPUs tornaram-se fundamentais para a implementação de Inteligência Artificial (IA) em larga escala e em nuvem, devido ao seu alto paralelismo de processamento e à superior largura de banda de memória.

Contudo, a implantação eficiente da IA em ambientes que transcendem os centros de dados equipados com infraestruturas de alto desempenho tornou-se um objetivo primordial tanto para a academia quanto para a indústria. Nesse cenário, embora as DNNs constituam a base para diversas tarefas de IA (DENG et al., 2020), há um interesse crescente em integrá-las a sistemas que operam em dispositivos com recursos limitados ou em aplicações que demandam processamento em tempo real. Segundo Menghani (2021), apesar do elevado desempenho desses modelos nas tarefas para as quais foram treinados, eles não são necessariamente eficientes para a implantação direta no mundo real, onde se exige a capacidade de tomar decisões sob demanda. Um desafio crítico reside na adequação do modelo às restrições do dispositivo de execução, visto que um elevado número de parâmetros durante a fase de inferência resulta em maior latência de processamento, além de ampliar o consumo de energia e a ocupação de memória em comparação a redes de menor escala.

Diante das limitações operacionais discutidas, as pesquisas de LeCun, Denker e Solla (1989) evidenciaram que as redes neurais artificiais, ou Artificial Neural Network (ANN) em inglês, são frequentemente superparamatrizadas, indicando que nem todos os seus parâmetros possuem relevância para o desempenho final. Esse cenário impulsionou o surgimento de vertentes de estudo dedicadas a reduzir o custo computacional e a necessidade de armazenamento das DNNs por meio da diminuição da quantidade de parâmetros e da construção de modelos de menor escala. Nesse sentido, Choudhary et al. (2020) ressaltam o crescente interesse e os progressos alcançados em técnicas de compressão, otimização e aceleração de modelos.

Para sistematizar esses avanços, Menghani (2021) desenvolveu um arcabouço conceitual que organiza os algoritmos e ferramentas voltados à eficiência em DL em cinco áreas fundamentais: técnicas de compressão, como quantização e poda; métodos de aprendizado, a exemplo da KD; estratégias de automação, que englobam a otimização de hiperparâmetros, ou HyperParameter Optimization (HPO) em inglês (FEURER; HUTTER, 2019), e a NAS, o desenvolvimento de arquiteturas eficientes, baseadas em camadas convolucionais e de atenção; e a infraestrutura, referente às ferramentas que auxiliam na implementação de modelos otimizados.

Complementando a discussão sobre a necessidade de modelos enxutos para ambientes reais, o estudo de Tabbakh et al. (2024) propõe um *framework* de IA verde, ou Green AI em inglês, que aborda tanto a infraestrutura de *hardware* quanto o refinamento algorítmico para viabilizar a sustentabilidade no desenvolvimento de IA. Os autores ressaltam que, apesar do papel central das GPUs na computação paralela, o seu consumo energético elevado, o calor gerado e os altos custos operacionais e de manutenção configuram obstáculos críticos para a escalabilidade e o acesso equitativo. Como alternativa para mitigar essas dependências, a pesquisa defende a transição para *hardwares* especializados, como unidades de processamento de tensores, ou Tensor Processing Unit (TPU) em inglês, e Field-Programmable Gate Arrays (FPGA), além do uso prospectivo de tecnologias emergentes, como a computação neuromórfica, que imita o cérebro para maior eficiência, e a computação quântica.

Alinhando-se aos eixos de compressão e arquiteturas eficientes discutidos por Menghani (2021), o trabalho detalha técnicas de poda estrutural e de magnitude, fundamentadas na Hipótese do Bilhete de Loteria, e de quantização para redução da precisão numérica e do uso de memória. No campo prático, as contribuições do estudo evidenciam que a destilação de conhecimento permite que modelos como o DistilBERT (SANH et al., 2020). Adicionalmente, em visão computacional, o emprego de convoluções separáveis em profundidade na arquitetura MobileNetV2 e a otimização via NAS, exemplificada pelo ProxylessNAS, demonstram reduções substanciais na complexidade computacional e no consumo de energia, viabilizando o processamento sob demanda em dispositivos de borda e internet das coisas, ou Internet of Things (IoT) em inglês.

2.3 Destilação de conhecimento

A eficiência computacional constitui um requisito crítico para a viabilização de modelos robustos em cenários de produção. Nesse contexto, a KD, conforme revisado por Gou et al. (2021), tem como objetivo primordial transferir o conhecimento de uma rede complexa e com elevada capacidade de generalização, denominada professor, para uma rede mais compacta e leve, o aluno. Essa abordagem permite que o modelo menor aprenda a mimetizar o comportamento do professor, reduzindo significativamente o número de parâmetros e a latência de inferência. A gênese dessa técnica remonta ao trabalho de Buciluǎ, Caruana e Niculescu-Mizil (2006), que investigaram a inviabilidade de empregar *ensembles* de alto desempenho em dispositivos com severas restrições de memória e processamento, como sensores e aparelhos auditivos.

Para mitigar essa limitação, os autores propuseram a compressão de modelos fundamentada na capacidade de aproximação universal das redes neurais, treinando o modelo aluno para replicar a função de predição de sistemas complexos a partir de grandes volumes de pseudo-dados rotulados pelo modelo original. Uma das contribuições centrais da pesquisa foi o algoritmo MUNGE, uma técnica não paramétrica para geração de dados sintéticos baseada na permutação estocástica de atributos entre vizinhos próximos, visando preservar a estrutura de densidade da distribuição original. Experimentos conduzidos em oito bases de dados de classificação binária demonstraram que as redes resultantes capturaram, em média, 97% do desempenho dos *ensembles*, sendo aproximadamente 1.000 vezes menores e mais velozes. Contudo, o estudo identifica que a eficácia do método é reduzida em problemas com atributos nominais de alta cardinalidade, além de constatar que a amostragem de dados reais não rotulados, quando disponível, supera em eficiência a geração sintética via MUNGE.

A técnica consolidou sua relevância a partir dos estudos de Ba e Caruana (2014) e Hinton, Vinyals e Dean (2015), que estabeleceram os fundamentos da KD. Em particular, Ba e Caruana (2014) investigaram se a profundidade é de fato um requisito intrínseco ao desempenho superior das redes neurais ou se a eficácia de arquiteturas profundas decorre majoritariamente de facilidades no processo de otimização. Para abordar essa questão, os autores utilizaram a compressão de modelos para treinar uma rede rasa, ou Shallow Neural Network (SNN) em inglês, visando imitar o comportamento de *ensembles*. A inovação metodológica central residiu no emprego da regressão de *logits* com perda $L2$, o que permitiu ao modelo aluno aprender a função de mapeamento interna do professor com maior fidelidade, evitando a perda de informação inerente ao espaço de probabilidades. Adicionalmente, foi introduzida uma camada linear de gargalo para decompor a matriz de pesos, acelerando a convergência de SNNs com elevado número de parâmetros.

Resultados experimentais nos *benchmarks* TIMIT e CIFAR-10 demonstraram que as SNNs podem atingir níveis de precisão comparáveis aos professores; no TIMIT, por exemplo, uma SNN mimetizada alcançou uma taxa de erro de fonemas de 20,0%, aproximando-se dos 19,5% obtidos por uma CNN. Concluiu-se que, embora a profundidade facilite o aprendizado

com algoritmos e métodos de regularização contemporâneos, SNNs possuem capacidade representacional suficiente para aprender funções complexas, desde que guiadas por um modelo professor. Entretanto, a abordagem apresenta limitações práticas, pois o sucesso do treinamento depende da disponibilidade de um professor de alta acurácia e, preferencialmente, de grandes volumes de dados não rotulados para assegurar a generalização.

Complementando as investigações de Ba e Caruana (2014), Hinton, Vinyals e Dean (2015) abordaram a inviabilidade computacional de implantar *ensembles* em dispositivos com restrições de latência e recursos, a despeito de sua reconhecida superioridade preditiva. Para solucionar esse impasse, os autores formalizaram o conceito de KD, utilizando as probabilidades de classe produzidas pelo professor como *soft labels*. A técnica distingue-se de abordagens convencionais ao introduzir um parâmetro de temperatura na função Softmax, o que suaviza a distribuição de saída e revela a estrutura de similaridade latente entre as classes, informações que seriam suprimidas em um treinamento tradicional baseado em *hard labels*.

Entre as principais contribuições teóricas, o trabalho demonstra que a minimização da diferença quadrática entre *logits* constitui um caso especial da destilação sob altas temperaturas, além de propor o uso de modelos especialistas treinados em paralelo para mitigar o custo de processamento em conjuntos de dados massivos, como o JFT. Resultados experimentais corroboram a eficácia da proposta em sistemas comerciais de reconhecimento de voz, nos quais um único modelo destilado alcançou uma acurácia de 60,8%, aproximando-se dos 61,1% obtidos por um *ensemble* de dez modelos e superando o modelo base treinado de forma isolada. Adicionalmente, a destilação demonstrou forte capacidade de regularização, permitindo que os modelos atingissem alto desempenho utilizando apenas 3% dos dados originais. Contudo, os autores apontam como limitação a dificuldade de consolidar o conhecimento de múltiplos especialistas em uma única rede de grande porte.

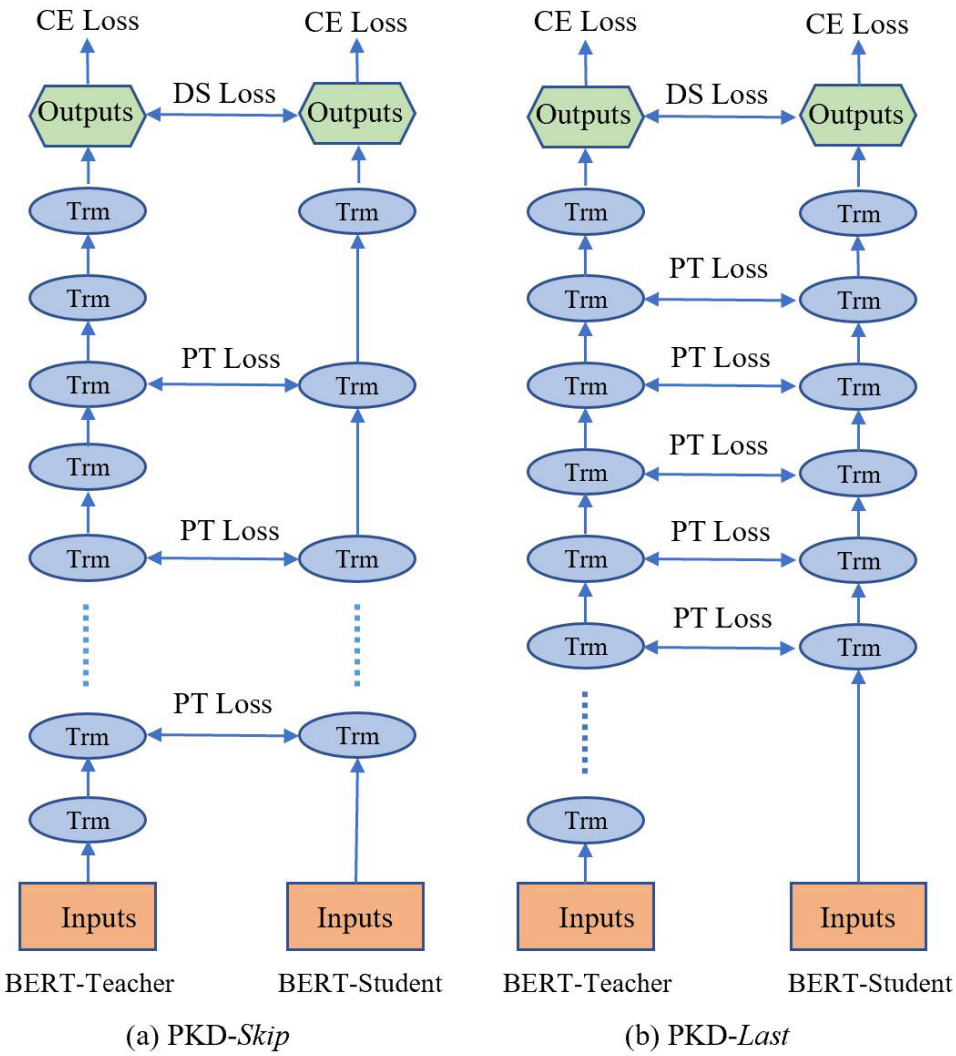
No âmbito de PLN, Gou et al. (2021) ressaltam que a KD é amplamente investigada para a concepção de modelos de linguagem leves e eficientes. Nesse cenário, destaca-se o trabalho de Sun et al. (2019), que abordou a elevada demanda de recursos computacionais e memória de modelos pré-treinados como o BERT, cuja aplicação é frequentemente dificultada em dispositivos de *hardware* limitado. Para mitigar esse desafio, os autores propuseram o método Patient Knowledge Distillation (Patient-KD), uma técnica de compressão que incentiva o modelo aluno a imitar, de forma incremental, as representações do *token* `[CLS]` das camadas intermediárias do professor, extrapolando o foco convencional na camada de saída.

A metodologia fundamenta-se em duas estratégias: a PKD-Last, que extrai conhecimento das últimas camadas, e a PKD-Skip, que realiza a destilação em camadas alternadas para capturar uma diversidade maior de semântica e abstração. Validações experimentais nos *benchmarks* GLUE e RACE revelaram que um modelo de seis camadas treinado via Patient-KD mantém a eficácia do original em conjuntos de dados volumosos, como SST-2 e MNLI, enquanto proporciona um aumento na velocidade de inferência de 1,94x e uma redução de 1,64x no uso

de memória. Todavia, o estudo aponta limitações em bases de dados reduzidas, onde há maior propensão ao sobreajuste, e discute o impacto do desalinhamento na inicialização dos pesos quando o aluno é derivado de um professor significativamente mais profundo sem um estágio de pré-treinamento específico.

A Figura 1 ilustra ambas as abordagens, onde "CE *loss*" representa a perda de entropia cruzada, ou Cross Entropy Loss (CE) em inglês, "DS *loss*" denota a função de custo que mensura a divergência entre as distribuições de probabilidade e "PT *loss*" indica o erro quadrático médio, ou Mean-Squared Error (MSE) em inglês, aplicado aos estados ocultos. Os termos "*Inputs*" e "*Outputs*" referem-se às entradas e saídas do sistema, respectivamente, enquanto "Trm" designa as camadas de arquitetura Transformers.

Figura 1 – Arquitetura do método Patient Knowledge Distillation aplicado ao BERT



Fonte: Sun et al. (2019).

Analogamente, Sanh et al. (2020) e Tang et al. (2019) também empregaram a KD para compactação da arquitetura BERT. Especificamente, Sanh et al. (2020) abordaram a elevada

complexidade computacional e os impactos ambientais de modelos de linguagem pré-treinados em larga escala, fatores que restringem sua implementação em dispositivos de borda ou em cenários com orçamentos limitados. Para mitigar esses obstáculos, os autores propuseram o DistilBERT, uma variante compacta e generalista que, diferentemente de propostas anteriores voltadas a tarefas específicas, utiliza a destilação durante a etapa de pré-treinamento.

A metodologia consistiu na redução de 50% do número de camadas do modelo original e na aplicação de uma função de perda tripla que integra as perdas de modelagem de linguagem, de destilação e de distância de cosseno, visando alinhar os estados ocultos do modelo aluno aos do professor. Os resultados experimentais indicaram que o DistilBERT preserva cerca de 97% da capacidade de compreensão de linguagem do BERT no *benchmark* GLUE, sendo 40% menor em termos de parâmetros e 60% mais veloz na inferência. Embora o estudo comprove a viabilidade do modelo para computação em tempo real em dispositivos móveis, os autores ressaltam que a estratégia de compressão priorizou a redução da profundidade da rede, fundamentada na premissa de que variações na dimensão oculta exercem menor impacto na eficiência computacional do que a diminuição do número de camadas.

Por sua vez, Tang et al. (2019) investigaram as limitações práticas decorrentes da profundidade de arquiteturas como o BERT e o GPT-2, cujos requisitos de latência dificultam a execução em dispositivos com recursos limitados. Para mitigar esse entrave, os autores propuseram uma metodologia de KD que transfere a competência de um professor para uma rede BiLSTM de camada única. A contribuição central do estudo reside no desenvolvimento de uma técnica de *data augmentation* baseada em heurísticas agnósticas à tarefa, incluindo mascaramento aleatório, substituição de termos guiada por *part-of-speech tags* e amostragem de *n-grams*, visando expandir o conjunto de transferência e otimizar o aprendizado a partir dos *logits* do professor. Resultados experimentais nos *benchmarks* SST-2, MNLI e QQP evidenciaram que o modelo destilado atinge resultados comparáveis ao ELMo, operando com cerca de 100 vezes menos parâmetros e apresentando uma inferência até 15 vezes mais célere. Entretanto, os autores observam que, embora a técnica reduza a disparidade entre redes recorrentes e Transformers profundos, a eficácia final ainda se situa entre 4 e 7 pontos percentuais abaixo dos modelos originais, o que reforça a existência de um compromisso necessário entre eficiência operacional e precisão preditiva.

Adhikari et al. (2020) investigaram os elevados custos computacionais e as demandas de memória associados ao uso de modelos baseados em Transformers, como o BERT, especialmente na classificação de documentos contendo sequências extensas. Para enfrentar esse desafio, os autores propuseram o emprego da KD utilizando uma variante refinada do modelo, denominada DocBERT, como professor para transferir conhecimento a arquiteturas significativamente mais simples e eficientes, como redes LSTM de camada única, redes neurais convolucionais e modelos de regressão logística. A metodologia fundamenta-se na minimização da divergência de Kullback-Leibler entre as probabilidades de saída do professor e do aluno, auxiliada por um

conjunto de transferência expandido via técnicas de aumento de dados, como substituição de palavras guiada por classes gramaticais e mascaramento aleatório.

Como principal contribuição, o estudo demonstra a viabilidade de destilar o conhecimento de modelos de linguagem complexos para modelos lineares, revelando que um modelo KD-LSTM pode atingir paridade de desempenho com o professor em *benchmarks* como Reuters, AAPD e IMDB, utilizando apenas 3% dos parâmetros e reduzindo em 40 vezes o volume de operações de ponto flutuante, ou Floating Point Operations (FLOPs) em inglês. Entretanto, os autores ressaltam que a destilação não reduz a carga computacional da fase de treinamento e apontam a sensibilidade do modelo ao truncamento de documentos, evidenciando que a redução excessiva da sequência máxima de entrada compromete a precisão em bases de dados compostas por textos longos.

A atratividade da KD reside em sua aparente generalidade, sugerindo que qualquer modelo aluno poderia, em princípio, aprender a partir de qualquer professor. No entanto, motivados por evidências de experimentos com resultados insatisfatórios, como a dificuldade observada por Zagoruyko e Komodakis (2017) em obter ganhos significativos no conjunto de dados ImageNet, Cho e Hariharan (2019) investigaram os fatores que levam a técnica a não aprimorar o desempenho da rede. As conclusões indicam que a KD não constitui uma solução universal, apresentando limitações quando a capacidade representacional do aluno é insuficiente para mimetizar o professor com êxito. Além disso, observou-se que interromper precocemente o treinamento do professor pode ser uma estratégia eficaz para mitigar o problema da disparidade de capacidade, facilitando a adaptação do aluno à função de perda. Diante dessas evidências, torna-se essencial determinar critérios para a escolha da arquitetura do estudante, assegurando que este seja capaz de integrar efetivamente o conhecimento transferido pelo professor.

2.4 Arquiteturas Mixture of Experts

À medida que as aplicações de IA se expandem, Mu e Lin (2025) ressaltam que a complexidade dos sistemas impõe dois desafios centrais: o custo computacional proibitivo para o treinamento e a implantação de modelos de grande porte, e a dificuldade de integrar conhecimentos heterogêneos e conflitantes em uma única arquitetura, o que frequentemente resulta em instabilidades e desempenho subótimo. Tais fatores evidenciam a necessidade de estruturas mais eficientes e escaláveis, como a arquitetura MoE. Baseada em uma estratégia de "dividir para conquistar", a MoE diferencia-se dos modelos densos tradicionais por não ativar a totalidade de seus parâmetros para cada entrada; em vez disso, o modelo seleciona dinamicamente apenas o subconjunto de parâmetros mais relevante com base nas características dos dados processados. Esse mecanismo de ativação seletiva favorece a especialização dos componentes e mitiga as dificuldades associadas ao aprendizado em cenários de tarefas diversas, permitindo expandir significativamente a capacidade do sistema sem um aumento proporcional

no custo computacional, o que assegura um equilíbrio entre desempenho e eficiência operacional, como afirmam Fedus, Zoph e Shazeer (2022).

Zuo et al. (2022) abordam os desafios da latência excessiva e do elevado número de parâmetros em modelos de linguagem pré-treinados, argumentando que métodos convencionais de compressão via KD podem comprometer a capacidade de representação do modelo. Para mitigar esse problema, os autores propõem o MoEBERT, uma arquitetura que integra a estrutura MoE ao processo de ajuste fino, visando aumentar tanto a capacidade quanto a velocidade de inferência. A metodologia consiste na adaptação das FNNs em múltiplos especialistas, empregando uma estratégia baseada na importância dos neurônios: os componentes com maiores pontuações são compartilhados entre os especialistas para preservar a representação original, enquanto os demais são distribuídos para promover a diversidade. O treinamento é conduzido por um algoritmo de KD camada a camada específico para a tarefa, permitindo a ativação de apenas um especialista por *token* durante a inferência, o que assegura um custo computacional estável. Resultados experimentais nos *benchmarks* GLUE e SQUAD demonstram que o MoEBERT supera algoritmos de destilação do estado da arte, alcançando melhorias de 2% no *dataset* MNLI e um incremento de 7,0 pontos em F1 no SQUAD v2.0, além de reduzir os parâmetros efetivos de 110M para 66M. Embora o modelo se mostre robusto a diferentes estratégias de roteamento, o estudo concentra-se na destilação específica para a tarefa, explorando a redundância dos modelos de linguagem para igualar ou superar o desempenho das versões integrais densas.

Além da aplicação em tarefas de compressão, as arquiteturas MoE têm sido exploradas para lidar com a heterogeneidade de dados em ambientes dinâmicos. Nesse contexto, Zhao et al. (2023) abordaram a detecção de notícias falsas em cenários multidomínio, ressaltando que modelos focados em domínios únicos apresentam desempenho limitado diante da diversidade temática das redes sociais contemporâneas. Para superar essa limitação, os autores propuseram um *framework* fundamentado em MoE que combina o processamento via BertTokenizer e o codificador de texto CLIP para extrair características de fusão robustas, minimizando ruídos e redundâncias informacionais. A inovação central reside no módulo de colaboração adaptativo, que ajusta a contribuição de especialistas baseados em TextCNN ao integrar análises de sentimento, *embeddings* de sentença baseados em atenção e informações de domínio. Resultados experimentais no *dataset* Weibo21 demonstraram que o modelo supera métodos consolidados, como MDFEND e EANN, alcançando um F1-score médio de 92,23%. Contudo, o estudo aponta limitações decorrentes do desbalanceamento entre categorias temáticas e da sensibilidade a ruídos inerentes ao ambiente digital, fatores que podem restringir a eficácia em domínios de alta complexidade técnica ou subjetividade.

Ainda no contexto de especialização em domínios técnicos, Lu, Liu e Nie (2025) investigaram os desafios da classificação automática de notícias financeiras, tarefa dificultada pela complexidade das estruturas linguísticas e pela terminologia altamente específica do setor.

Para enfrentar essas limitações, os autores propuseram o emprego do modelo Qwen1.5-MoE-A2.7B, fundamentado na arquitetura MoE, que utiliza apenas 2,7 bilhões de parâmetros ativados para rivalizar com modelos densos significativamente maiores. A abordagem metodológica destaca-se pelo uso da técnica Quantized Low-Rank Adaptation (QLoRA) para o ajuste fino instrucional, empregando quantização NormalFloat de 4 bits e otimizadores paginados, o que viabiliza o treinamento eficiente mesmo sob restrições severas de memória computacional.

Entre as inovações arquiteturais, o trabalho introduz o conceito de *fine-grained experts*, expandindo o sistema para 64 especialistas por meio da fragmentação de camadas de FNNs, além de implementar um mecanismo de roteamento aprimorado e uma técnica de inicialização por *upcycling*. Os resultados experimentais, obtidos a partir de notícias financeiras coletadas no Twitter, demonstram que o modelo atingiu uma acurácia de 89,10%, superando o desempenho de arquiteturas consolidadas como BERT, RoBERTa e ChatGLM. Adicionalmente, a estrutura MoE proporcionou uma redução de 75% nos custos de treinamento e um incremento de 1,74 vezes na velocidade de inferência em relação ao modelo denso Qwen1.5-7B, evidenciando elevada eficiência no emprego de recursos.

Consolidando o uso de arquiteturas esparsas para modelos de linguagem de larga escala, Fedus, Zoph e Shazeer (2022) investigaram os desafios de escalabilidade e os elevados custos computacionais de modelos densos, propondo o Switch Transformer como uma alternativa fundamentada no paradigma MoE. A inovação metodológica central reside na simplificação do algoritmo de roteamento, que direciona cada *token* a um único especialista, reduzindo drasticamente a carga de comunicação e a complexidade de cálculo sem comprometer a qualidade das representações. Para assegurar a estabilidade no treinamento de modelos que alcançam 1,6 trilhão de parâmetros, os autores introduziram o uso de precisão seletiva, empregando float32 nas operações do roteador para mitigar instabilidades numéricas típicas do formato bfloat16, além de esquemas de inicialização de pesos reduzidos e a técnica de *expert dropout* para prevenir o sobreajuste durante o ajuste fino.

Experimentalmente, o Switch Transformer demonstrou eficiência superior ao atingir velocidades de pré-treinamento até sete vezes maiores que o modelo T5-Base sob o mesmo orçamento de FLOPs, apresentando ganhos consistentes em *benchmarks* como GLUE, SuperGLUE e tarefas de resposta a perguntas em contextos multilíngues. Adicionalmente, o estudo explorou a viabilidade da compressão via KD, na qual modelos densos miniaturizados retiveram cerca de 30% dos ganhos de qualidade de seus professores esparsos massivos, a despeito de uma redução de até 99% no número de parâmetros. Contudo, os autores apontam como limitações a persistência de instabilidades em modelos de escala extrema, como a variante Switch-XXL, e a complexa relação entre a melhoria da perplexidade no pré-treinamento e a transferência efetiva de desempenho para tarefas de raciocínio lógico.

3 Materiais e Métodos

Este capítulo descreve a metodologia experimental desenvolvida com o propósito de validar a hipótese de que arquiteturas neurais compactas constituem uma alternativa viável aos modelos massivos na tarefa de classificação de BDPs. A abordagem proposta transcende a avaliação convencional baseada apenas na eficácia preditiva ao integrar métricas de sustentabilidade e eficiência computacional, visando responder aos desafios inerentes à implementação em cenários industriais reais.

A estrutura metodológica organiza-se em quatro eixos fundamentais. Inicialmente, a seção 3.1 detalha os conjuntos de dados utilizados, estabelecendo a distinção entre os *corpora* de domínio geral, empregados no pré-treinamento continuado, e os dados proprietários da indústria de petróleo e gás, destinados ao ajuste fino. Na sequência, a seção 3.2 define os critérios de avaliação, os quais combinam métricas clássicas de classificação, como F1-Score, Precisão e Revocação, com indicadores de eficiência alinhados aos princípios de Green AI, incluindo o consumo energético, a pegada de carbono e o custo computacional.

Em continuidade, a seção 3.3 apresenta o cerne da proposta, descrevendo as arquiteturas neurais selecionadas, que abrangem desde modelos fundamentais, como a CNN e a BiLSTM, até abordagens baseadas em BERT e MoE, além das técnicas de compressão aplicadas, com ênfase na KD. Por fim, a seção 3.4 define o protocolo experimental, detalhando a infraestrutura de *hardware*, as configurações de hiperparâmetros e os métodos estatísticos adotados para assegurar a reprodutibilidade e a validade dos resultados comparativos.

3.1 Bases de dados

A eficácia das arquiteturas neurais propostas depende intrinsecamente da qualidade e da representatividade dos dados utilizados nas diferentes etapas de aprendizagem. Esta seção apresenta uma visão geral dos recursos textuais e das estratégias de processamento adotadas no estudo. Inicialmente, a subseção 3.1.1 detalha a estratégia de treinamento continuado destinada à atualização dos parâmetros de modelos de linguagem pré-treinados, ou Pre-trained Language Model (PLM) em inglês, descrevendo os corpora não supervisionados empregados para a DAPT e a adaptação voltada à tarefa, ou Task-Adaptive Pre-Training (TAPT) em inglês. De forma complementar, a subseção 3.1.2 aborda o conjunto de dados proprietário de Boletins Diários de Perfuração (BDPs) e a adaptação dessas redes para a tarefa específica de classificação. Por fim, a subseção 3.1.3 descreve os procedimentos de preparação dos dados, processo que engloba o tratamento de inconsistências, o pré-processamento textual e a estruturação necessária para a realização dos experimentos.

3.1.1 Treinamento continuado

O treinamento continuado constitui uma estratégia para a atualização de modelos de linguagem, na qual a rede é submetida a épocas adicionais de ajuste utilizando um novo conjunto de textos, distintos dos empregados no pré-treinamento inicial, mas preservando o objetivo de aprendizado original. Sendo um processo auto-supervisionado, essa técnica aproveita a abundância de dados não rotulados, priorizando o uso de um *corpus* composto por documentos integrais em detrimento de sentenças isoladas. Tal preferência justifica-se pela possibilidade de criar sequências de múltiplas sentenças contíguas, o que permite ao modelo capturar dependências de longo alcance e representações contextuais mais ricas.

Nesse contexto, o treinamento continuado é tradicionalmente segmentado em duas vertentes: a TAPT e a DAPT. O TAPT fundamenta-se no emprego de textos não rotulados vinculados a uma tarefa específica, como a análise de sentimentos, visando o refinamento das representações para um propósito determinado. Por sua vez, o DAPT tem como objetivo adequar o modelo às particularidades de um domínio de conhecimento especializado, a exemplo do setor de petróleo e gás, permitindo a captura de nuances terminológicas e semânticas características desse campo de atuação.

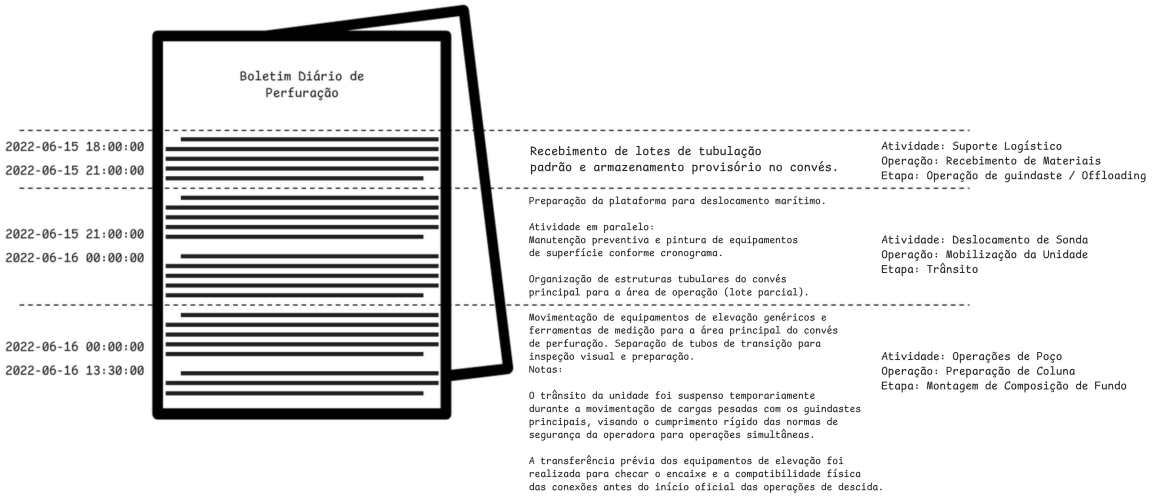
Nesta etapa, utilizou-se duas bases de dados que satisfazem as propriedades requeridas, cenário no qual se aplica a técnica DAPT para a adaptação dos modelos ao português brasileiro e o TAPT para a tarefa de classificação de BDPs. O primeiro conjunto de dados consiste no brWaC (FILHO et al., 2018), um extenso *corpus* multidomínio em português brasileiro extraído da *web*, composto por 2,68 bilhões de *tokens*. Conforme definido por Caseli e Nunes (2024), *tokens* constituem sequências de caracteres às quais se atribui um valor, podendo representar palavras, subpalavras ou caracteres individuais. Tais elementos encontram-se distribuídos em $3{,}53$ milhões de documentos provenientes de mais de 120.000 endereços eletrônicos, o que consolida o brWaC como o maior *corpus* aberto da língua portuguesa. Além de sua escala, a base contempla documentos íntegros, característica que assegura alta diversidade temática e qualidade de conteúdo, em consonância com sua metodologia de construção.

A segunda base de dados, por sua vez, compreende um conjunto de 808.878 descrições de BDPs extraídas de 302 poços, consistindo em sentenças não rotuladas em texto livre que detalham o processo de perfuração. Esses registros, fornecidos pela Petrobras, abrangem um volume expressivo de informações e possuem caráter restrito, não estando disponíveis publicamente. Os BDPs relatam as atividades conduzidas durante intervenções em poços, conjuntos de operações destinadas à construção, manutenção e abandono seguro de poços de petróleo ao longo de seu ciclo de vida. Cada intervenção é pautada por objetivos específicos e requer um projeto executivo acompanhado de estimativas de custos, estando sujeita a interrupções até a sua efetiva conclusão.

A Figura 2 ilustra a estrutura e o conteúdo típicos desses documentos, evidenciando

como os eventos operacionais são registrados cronologicamente através de estampas de tempo, ou *timestamps* em inglês, associadas a uma descrição narrativa em linguagem natural. Além do detalhamento textual, a figura apresenta a estrutura de metadados utilizada para a classificação técnica, dividida nos níveis de Atividade, Operação e Etapa, o que permite a categorização hierárquica das intervenções.

Figura 2 – Exemplificação da estrutura de um BDP.



Fonte: Elaborada pelo autor.

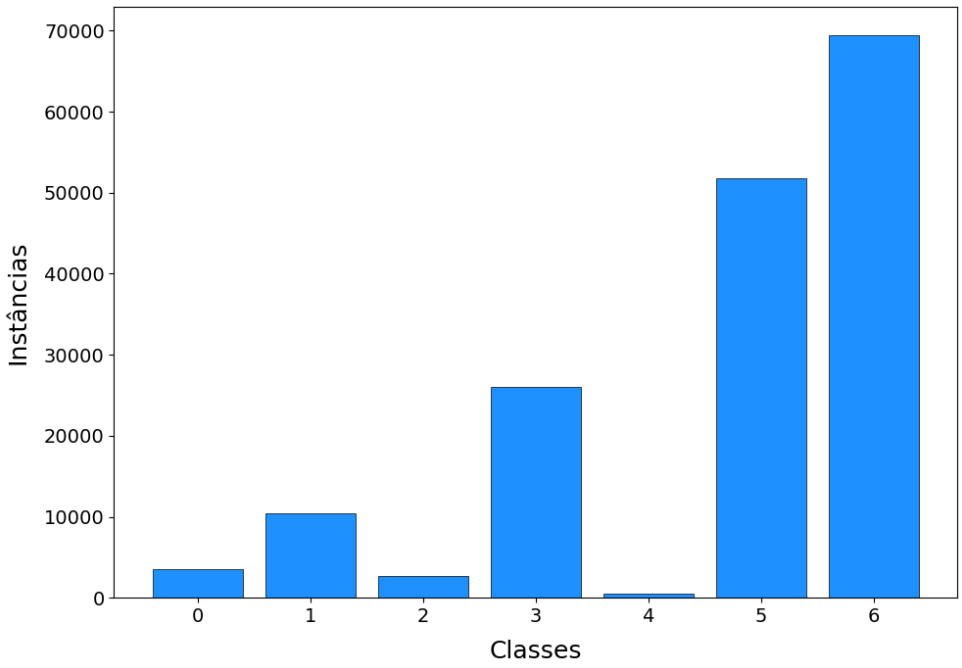
Tais intervenções são classificadas em quatro categorias fundamentais, a saber: perfuração, completação, avaliação exploratória e *workover*. O presente trabalho delimita seu escopo à categoria de perfuração, a qual compreende operações voltadas ao alcance de objetivos geológicos, à avaliação de poços exploratórios e à instalação de equipamentos necessários para a produção de hidrocarbonetos, bem como para a injeção de fluidos no reservatório em poços de desenvolvimento.

3.1.2 Ajuste fino

Complementarmente ao treinamento continuado, o ajuste fino constitui outro método fundamental para a adaptação de modelos de linguagem. Nesse processo, utilizam-se dados rotulados e uma função objetivo específica para a tarefa-alvo, como a CE em problemas de classificação. De modo análogo ao treinamento continuado, o ajuste fino promove a atualização dos pesos da rede. Contudo, por ser uma etapa direcionada e concentrada na tarefa final, requer usualmente um volume menor de dados e resulta em um modelo altamente especializado no contexto de aplicação pretendido.

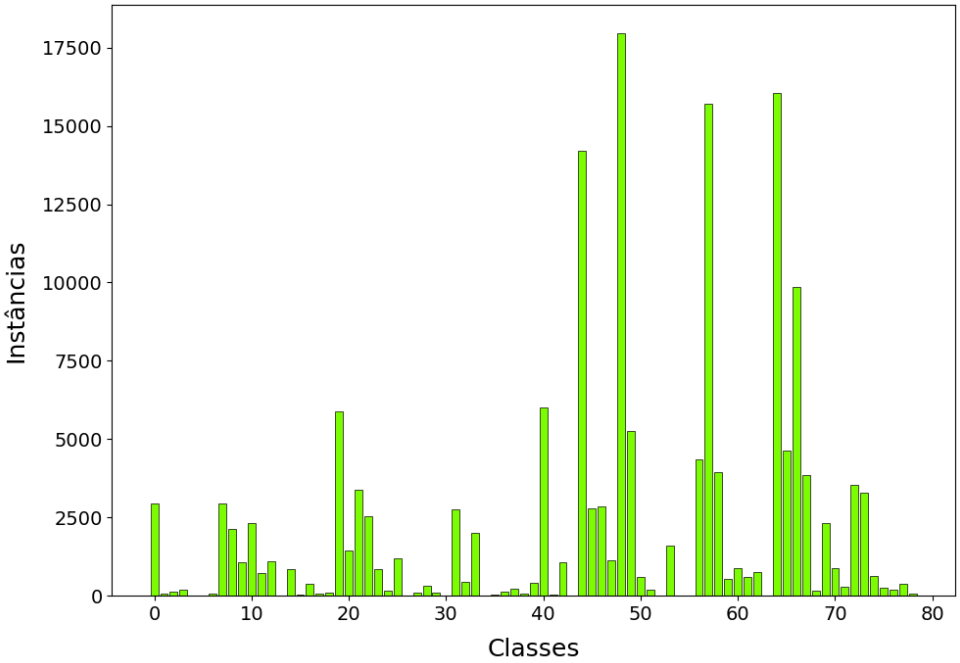
Para a referida etapa de ajuste fino, utilizou-se um segundo conjunto composto por 205.261 descrições de BDPs, extraídas de 307 poços de petróleo perfurados nos últimos cinco anos e rotuladas especificamente para a tarefa de classificação de sentenças. Esses registros foram estruturados em três níveis de uma hierarquia multirrótulo, organizados em ordem crescente de especificidade: (i) Atividade; (ii) Operação; e (iii) Etapa. O nível de Atividade, constituído por sete classes, representa o estrato de menor complexidade e, apesar do desbalanceamento dos dados, apresenta distinções semânticas que favorecem a classificação. Em contrapartida, os níveis de Operação e Etapa impõem desafios superiores devido à elevada cardinalidade das classes, 79 e 254, respectivamente, ao acentuado desequilíbrio na distribuição das amostras e à similaridade textual entre categorias distintas. As Figura 3, Figura 4 e Figura 5 ilustram a distribuição das classes em cada um desses níveis. Ressalta-se que, de forma análoga ao *corpus* empregado no treinamento continuado, esta base de dados possui caráter privado.

Figura 3 – Distribuição de classes para o nível de Atividade.



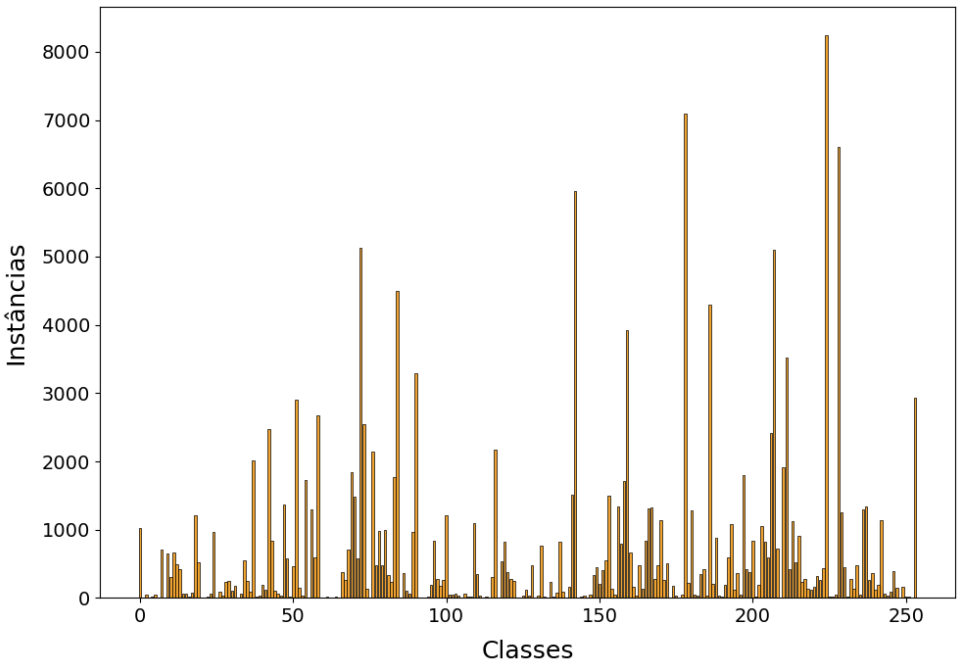
Fonte: Elaborada pelo autor.

Figura 4 – Distribuição de classes para o nível de Operação.



Fonte: Elaborada pelo autor.

Figura 5 – Distribuição de classes para o nível de Etapa.



Fonte: Elaborada pelo autor.

Diante dessa estruturação hierárquica, nota-se uma disparidade significativa no volume de instâncias entre as classes do problema. No nível de Atividade, especificamente, a classe seis concentra um pouco menos da metade das amostras do conjunto de dados, ao passo que

a classe quatro é extremamente subrepresentada, com muito menos de 10.000. Esse cenário de desbalanceamento acentua-se nos estratos mais granulares: no nível de Operação, apenas cinco classes majoritárias, dentre as 79 catalogadas, abrangem cerca de 45% das instâncias, embora a representatividade ideal para um conjunto equilibrado devesse ser de aproximadamente $1,27\%$ por categoria. De modo análogo, no nível de Etapa, nove classes predominantes, em um universo de 254, respondem por cerca de 31% das amostras, distanciando-se da proporção teórica de $0,39\%$ necessária para uma distribuição uniforme.

3.1.3 Preparação dos dados

Para assegurar a integridade dos dados, o processo de preparação envolveu, primordialmente, a eliminação de registros duplicados e a normalização dos textos. Em consonância com as diretrizes estabelecidas por Souza, Nogueira e Lotufo (2020a), utilizou-se exclusivamente o corpo dos documentos da base brWaC e desconsideraram-se os títulos. O processamento desses textos contou com o auxílio da biblioteca ftfy (SPEER, 2019) para a remoção de *tags* HTML residuais e a correção de *mojibakes*, distorções de caracteres resultantes de falhas na decodificação, procedimento este estendido aos demais conjuntos de dados. Adicionalmente, as bases de BDPs foram submetidas a uma etapa de tratamento específica, fundamentada em algoritmos heurísticos desenvolvidos pelos pesquisadores do projeto ProtoRADIAR. Tal fase visou à correção de erros ortográficos e gramaticais frequentes nas descrições, além da remoção de ruídos textuais irrelevantes para a tarefa de classificação.

3.2 Métricas de avaliação

Com a finalidade de quantificar o desempenho dos modelos, selecionaram-se três métricas referentes à eficácia preditiva e cinco indicadores voltados à avaliação da eficiência computacional e da pegada energética. Nesse contexto, a subseção 3.2.1 detalha os índices empregados na tarefa de classificação, enquanto a subseção 3.2.2 descreve as medidas destinadas à mensuração da eficiência das arquiteturas. Fundamentado nesses critérios, busca-se o desenvolvimento de uma solução robusta capaz de maximizar a eficácia preditiva e, concomitantemente, minimizar a demanda por recursos, preservando-se a simplicidade arquitetural proposta.

3.2.1 Métricas para classificação

Para a avaliação das predições dos modelos, adotaram-se as métricas de precisão, ou *precision* em inglês, revocação, ou *recall* em inglês, e *F1-score*, as quais fornecem diferentes perspectivas para a quantificação da eficácia dos classificadores. Embora as formulações originais dessas métricas sejam voltadas para a classificação binária, fundamentando-se na distinção entre exemplos positivos e negativos, é possível generalizá-las para problemas multiclasse. Essa generalização pode ser efetuada tanto pelo cálculo individualizado por categoria quanto

pela aplicação de médias que consolidam os resultados em um único valor representativo do desempenho global. No presente estudo, adotou-se a estratégia baseada em médias para a avaliação das arquiteturas propostas.

3.2.1.1 Precisão

A precisão representa a proporção de amostras corretamente atribuídas a uma determinada classe em relação ao total de instâncias que o modelo classificou como pertencentes a essa categoria. Sob uma perspectiva intuitiva, essa métrica reflete a capacidade do classificador em mitigar a ocorrência de Falsos Positivos (FP), ou seja, em evitar que amostras de outras classes sejam incorretamente rotuladas como pertencentes à classe em análise. Formalmente, a precisão é calculada por meio da Equação 1:

$$\begin{aligned} \mathsf{P} &= \frac{1}{C}\sum_{c=1}^{C}\mathsf{p}_c \\ \mathsf{p}_c &= \frac{\mathsf{VP}_c}{\mathsf{VP}_c + \mathsf{FP}_c} \end{aligned} \tag{1}$$

onde C representa o número total de classes e p_c denota a precisão referente à classe c. O termo VP_c corresponde à quantidade de Verdadeiros Positivos (VP), definidos como as amostras positivas classificadas corretamente, enquanto FP_c indica o número de FPs, os quais consistem nas amostras negativas incorretamente categorizadas como positivas. O resultado dessa métrica varia no intervalo de zero a um, sendo que valores mais próximos da unidade sinalizam um desempenho superior.

3.2.1.2 Revocação

De modo complementar, a revocação exprime a proporção de instâncias corretamente classificadas em relação ao total de exemplos efetivamente pertencentes a uma determinada classe no conjunto de dados. Sob uma perspectiva intuitiva, essa métrica reflete a capacidade do modelo em identificar a totalidade das amostras positivas disponíveis e minimizar a incidência de Falsos Negativos (FN), o que consiste em evitar que amostras da classe em análise sejam incorretamente rotuladas como pertencentes a outras categorias. O cálculo da revocação é formalizado pela Equação 2:

$$\begin{aligned} \mathsf{R} &= \frac{1}{C}\sum_{c=1}^{C}\mathsf{r}_c, \\ \mathsf{r}_c &= \frac{\mathsf{VP}_c}{\mathsf{VP}_c + \mathsf{FN}_c}, \end{aligned} \tag{2}$$

onde C representa o número total de classes e r_c denota a revocação referente à classe c. O termo VP_c expressa a quantidade de VPs, ao passo que FN_c indica o número de FNs, correspondendo às amostras positivas classificadas incorretamente como negativas. Os valores

resultantes dessa métrica situam-se no intervalo de zero a um, sendo que índices mais elevados refletem um desempenho superior.

3.2.1.3 F1-Score

Finalmente, o F1-Score constitui uma métrica que consolida a precisão e a revocação, sendo definido pela média harmônica entre ambas, conforme apresentado na Equação 3:

$$\begin{aligned} \mathsf{F1} &= \frac{1}{C}\sum_{c=1}^{C} \mathsf{f1}_c, \\ \mathsf{f1}_c &= \frac{2 \times \mathsf{p}_c \times \mathsf{r}_c}{\mathsf{p}_c + \mathsf{r}_c}, \end{aligned} \tag{3}$$

onde C constitui o número total de classes e $f1_c$ denota o F1-Score referente à classe c. As variáveis p_c e r_c representam, respectivamente, a precisão e a revocação alcançadas pela mesma classe. Observa-se que a contribuição relativa de ambas as medidas para a composição do F1-Score é equivalente. Os valores possíveis para essa métrica variam no intervalo de zero a um, cenário em que índices mais elevados indicam um desempenho superior.

3.2.2 Métricas de eficiência

Sob uma perspectiva suplementar, avaliou-se o desempenho das arquiteturas em relação ao custo computacional por meio das métricas de consumo de energia e intensidade de carbono, aferidas pela ferramenta CodeCarbon, em conjunto com os indicadores de FLOPs, tempo de inferência e número de parâmetros. Para mensurar o esforço computacional demandado por um modelo, torna-se essencial a utilização de métricas que viabilizem uma comparação equitativa entre diferentes topologias de redes neurais. Esses indicadores devem, idealmente, apresentar estabilidade frente a distintas configurações experimentais e manter-se independentes do *hardware* utilizado na execução. Nesse contexto, a seleção das medidas fundamentou-se na análise proposta por Schwartz et al. (2020), com o objetivo de assegurar a padronização e a transparência na avaliação da eficiência dos modelos desenvolvidos.

3.2.2.1 Consumo de energia

O consumo de energia refere-se à estimativa da potência elétrica demandada pelo *hardware* subjacente durante a execução de um experimento computacional. O CodeCarbon (COURTY et al., 2024) monitora o fornecimento de energia aos principais componentes de *hardware* em intervalos regulares de 15 segundos. Esse procedimento permite capturar as variações temporais no consumo ao longo da execução dos modelos, abrangendo a GPU, a CPU e a memória RAM, cujas aferições seguem estratégias distintas condicionadas às limitações de acesso ao *hardware* e ao sistema operacional.

No que tange às GPUs NVIDIA, o monitoramento é realizado diretamente pela biblioteca nvidia-ml-py, que acessa as métricas de uso energético expostas pela API NVIDIA Management Library (NVML). Tal abordagem viabiliza uma medição precisa durante os experimentos, aspecto fundamental em cenários de treinamento e inferência de modelos de *deep learning*, nos quais a GPU atua frequentemente como o principal componente consumidor de energia.

Para a memória RAM, adota-se um modelo de estimativa fundamentado na quantidade de *slots* de memória utilizados, o qual assume um consumo base de 5 Watts por módulo DIMM em sistemas x86 e de $1{,}5$ Watts por módulo em arquiteturas ARM. O modelo impõe ainda limites mínimos de consumo, estabelecendo um piso de 10 Watts para sistemas x86, sob a premissa do uso de dois DIMMs, e de 3 Watts para sistemas ARM, a exemplo de dispositivos embarcados como o Raspberry Pi.

Por fim, a estimativa do consumo da CPU varia conforme o sistema operacional e a arquitetura do processador. Em ambientes Linux, sistema adotado neste trabalho, o consumo de processadores Intel e AMD é obtido via interface Intel Running Average Power Limit (RAPL), mediante o acesso a arquivos localizados no diretório /sys/class/powercap/intel-rapl/. Nesse processo, monitoram-se todos os pacotes de CPU disponíveis, desde que haja suporte do sistema à interface RAPL e as devidas permissões de leitura.

3.2.2.2 Intensidade de carbono

A intensidade de carbono corresponde à quantidade de dióxido de carbono equivalente, simbolizado por CO2e, emitida por unidade de energia elétrica consumida. Tal grandeza, usualmente expressa em gCO2e/kWh ou kgCO_2e/MWh, revela-se fundamental para a estimativa do impacto ambiental de experimentos computacionais, visto que relaciona diretamente o consumo energético do *hardware* às emissões de gases de efeito estufa associadas à geração da eletricidade utilizada.

No âmbito da ferramenta CodeCarbon, utilizam-se valores agregados de intensidade de carbono da eletricidade, classificados por país ou provedor de nuvem e obtidos a partir de bases de dados consolidadas, como a Our World in Data. Tal estratégia permite capturar as emissões médias vinculadas à geração de energia elétrica em diferentes regiões geográficas. Ressalta-se que, diante da indisponibilidade de dados do ano corrente para um determinado país, adota-se o valor referente ao ano mais recente como aproximação. Já nas situações em que inexistem informações sobre a intensidade de carbono ou a matriz energética nacional, o CodeCarbon emprega um valor médio global estimado em 475 gCO_2e/kWh, medida que visa assegurar a continuidade da estimativa de emissões.

3.2.2.3 Operações de ponto flutuante

As FLOPs oferecem uma estimativa do esforço computacional despendido, sendo mensuradas analiticamente por meio do custo de duas operações fundamentais em nível de

máquina: a adição, ou ADD, e a multiplicação, ou MUL. Com base nesses componentes primários, o custo de FLOPs de qualquer operação abstrata de DL, a exemplo da tangente hiperbólica, de multiplicações de matrizes e de convoluções, pode ser computado como uma função recursiva dessas operações de base. Conquanto o cálculo detalhado varie conforme a arquitetura específica do modelo, a métrica de FLOPs pode ser expressa de forma genérica pela Equação 4:

$$\text{FLOPs} = \sum_{l=1}^{L} \text{Q}_l, \tag{4}$$

onde L denota o número de camadas da rede e Q_l corresponde à quantidade de operações ADD e MUL realizadas pelo modelo na camada l.

De acordo com Schwartz et al. (2020), as FLOPs apresentam atributos favoráveis à avaliação de modelos, destacando-se por mensurarem diretamente o volume de processamento executado pelo sistema em uma instância específica da rede, o que estabelece uma relação direta com o consumo energético. Além disso, essa métrica é independente do *hardware* empregado, fator que possibilita comparações equitativas entre diferentes abordagens, e demonstra, usualmente, uma correlação significativa com o tempo de execução. Diante de tais propriedades, a FLOPs consolida-se como um indicador objetivo para a análise da eficiência de arquiteturas de DL.

3.2.2.4 Tempo de inferência

O tempo de inferência é definido como o intervalo total necessário para que um modelo processe uma entrada e gere o respectivo resultado. Trata-se de uma métrica intuitiva, pois, sob condições constantes, uma rede mais veloz tende a realizar um menor volume de trabalho computacional. Entretanto, Schwartz et al. (2020) advertem que essa medida é fortemente influenciada por fatores externos, como as especificações do *hardware* subjacente, a carga de outros processos em execução simultânea e o número de núcleos utilizados. Essas variáveis dificultam a comparação equitativa entre diferentes modelos e a distinção entre as contribuições da arquitetura e os ganhos de desempenho derivados do *hardware*. O cálculo do tempo de inferência é formalizado na Equação 5:

$$T_{\text{Inferência}} = \frac{\text{FLOPs}}{\text{FLOPS}}, \tag{5}$$

onde FLOPs refere-se ao total de operações de ponto flutuante realizadas pela rede, ao passo que FLOPS designa a quantidade de operações de ponto flutuante por segundo, ou Floating Point Operations per Second em inglês. Tal métrica reflete a velocidade de computação do *hardware* empregado no processamento.

3.2.2.5 Número de parâmetros

Por fim, a contagem de parâmetros constitui um critério adicional para a avaliação da eficiência e engloba o total de parâmetros, considerando tanto os congelados quanto aqueles passíveis de aprendizado. De modo análogo aos FLOPs e diferentemente do tempo de inferência, essa métrica independe do *hardware* empregado e apresenta correlação direta com o consumo de memória da rede. Contudo, conforme salientado por Schwartz et al. (2020), modelos que possuem quantidade similar de parâmetros podem realizar volumes distintos de trabalho computacional, visto que a forma como os algoritmos utilizam tais componentes varia substancialmente entre diferentes arquiteturas.

3.3 Abordagem proposta

O presente trabalho propõe a classificação das descrições presentes nos BDPs mediante a utilização de diferentes arquiteturas neurais, com ênfase na redução do número de parâmetros e da complexidade estrutural por intermédio de técnicas de compressão de modelos. Essa abordagem objetiva preservar o desempenho das soluções vigentes na literatura e, simultaneamente, otimizar a eficiência computacional do sistema. Em consonância com os experimentos de Rodrigues et al. (2022), autor que restringiu a análise ao nível de Etapa no contexto do problema, esta pesquisa trata a tarefa como três problemas de classificação distintos, sendo um para cada nível taxonômico descrito na seção 3.1. Dessa maneira, a predição das categorias ocorre de forma independente para cada estrato da hierarquia. O método proposto organiza-se em três etapas fundamentais, cujo fluxo de processamento encontra-se ilustrado no *pipeline* da Figura 6.

Figura 6 – *Pipeline* da metodologia proposta.

Fonte: Elaborada pelo autor.

3.3.1 Escolha das arquiteturas neurais

Inicialmente, a seleção das arquiteturas neurais para os experimentos fundamentou-se nos modelos propostos por Rodrigues et al. (2022) e Tang et al. (2019), bem como nas diretrizes extraídas da revisão sobre modelos de DL para classificação de textos de Minaee et al. (2021). Tal etapa culminou na definição de doze modelos baseados em cinco arquiteturas distintas, cujas características são detalhadas a seguir.

3.3.1.1 Convolutional Neural Network

As CNNs constituem uma das arquiteturas mais proeminentes no cenário atual da IA, destacando-se pela eficácia no processamento de dados com estrutura de grade, tais como imagens, sinais de áudio e, no âmbito do PLN, sequências textuais. Tais modelos fundamentam-se em uma organização hierárquica de camadas que propicia a extração automática de atributos, evoluindo de padrões elementares nas camadas iniciais para representações semânticas complexas nos níveis mais profundos. O diferencial fundamental das CNNs reside na capacidade de capturar dependências espaciais e padrões locais mediante o compartilhamento de pesos e a invariância à translação, características que asseguram tanto a eficiência computacional quanto a robustez das representações geradas a partir das sequências de entrada (KRICHEN, 2023).

A operação fundamental que sustenta o processamento nessas redes é a convolução, a qual, conforme detalhado por Krichen (2023), pode ser representada matematicamente para sinais discretos de acordo com a Equação 6:

$$(f * g)(n) = \sum_{m=-\infty}^{\infty} f(m) \times g(n-m) \times \Delta m \tag{6}$$

onde $(f * g)(n)$ denota o resultado da operação de convolução no índice n, valor que representa a saída obtida pela combinação ponderada entre o sinal de entrada e o filtro. A função $f(m)$ corresponde ao sinal de entrada avaliado no índice de soma m, variável que percorre o domínio dos dados, ao passo que $g(n-m)$ refere-se ao filtro deslocado em relação ao índice de saída n, o que viabiliza a extração de padrões locais. O termo Δm indica o passo de discretização entre amostras consecutivas.

A arquitetura típica de uma CNN fundamenta-se na alternância entre camadas convolucionais, onde os filtros mencionados extraem padrões locais, e camadas de *pooling*, voltadas à redução da dimensionalidade espacial das representações. Esse arranjo estrutural confere robustez frente a pequenas distorções nos dados e diminui o custo computacional do modelo. Para introduzir não linearidade e viabilizar o aprendizado de funções complexas, aplicam-se funções de ativação, dentre as quais a Unidade Linear Retificada, ou Rectified Linear Unit (ReLU) em inglês, definida matematicamente por $f(x) = \max(0, x)$, destaca-se como a mais utilizada por sua eficácia em mitigar o problema do desaparecimento do gradiente (KRICHEN, 2023).

A arquitetura da CNN adotada neste trabalho se inspira na TextCNN (KIM, 2014) e integra uma camada de *embeddings*, dois fluxos paralelos de convoluções, que simulam o comportamento bidirecional de modelos do estado da arte em tarefas de PLN, e um classificador FNN. Inicialmente, a camada de *embeddings* funciona como uma tabela de consulta que mapeia cada *token* do vocabulário em um vetor denso de 256 dimensões, reservando o índice zero para o *padding*. Subsequentemente, esses vetores são reorganizados para alimentar as camadas convolucionais. No fluxo direto, os *embeddings* são processados por um conjunto de camadas

convolucionais 2D, cada uma com 100 filtros e *kernels* de tamanhos 3, 4 e 5 ao longo da dimensão temporal, visando capturar padrões locais de diferentes extensões, de forma análoga a *n-grams*. Após a aplicação da função de ativação ReLU, utiliza-se o *global max pooling* na dimensão temporal para condensar cada mapa de características em um valor escalar único.

De maneira complementar, no fluxo reverso, a sequência de *embeddings* é invertida e processada por um conjunto idêntico de camadas convolucionais, o que permite a captura de padrões na ordem inversa e a incorporação de informação bidirecional sem o emprego de camadas recorrentes. As representações resultantes de ambos os fluxos são concatenadas em um vetor global de características e encaminhadas ao classificador FNN. Este estágio final é composto por uma camada de *dropout* e uma projeção linear para o número de classes, finalizada pela função de unidade linear de erro gaussiano, ou Gaussian Error Linear Unit (GELU) em inglês, cujas saídas correspondem aos *logits* utilizados no processo de classificação.

3.3.1.2 Bidirectional Long Short-Term Memory

As LSTMs figuram entre as primeiras e mais eficazes abordagens para o tratamento do problema do desaparecimento do gradiente, característica que as consolidou em diversas aplicações de PLN (SOUZA; NOGUEIRA; LOTUFO, 2020b). Embora apresentem similaridades estruturais com as RNNs convencionais, esses modelos distinguem-se pela substituição da unidade recorrente comum por uma célula de memória capaz de preservar um estado interno. Tal configuração arquitetural incorpora uma conexão recorrente com peso fixo unitário, a qual assegura a transmissão do gradiente ao longo de sucessivas etapas do treinamento, evitando sua dissipação ou explosão.

Conforme observado por Zhang et al. (2023a), as RNNs possuem memória de longo prazo consolidada em seus parâmetros, que assimilam o conhecimento geral dos dados durante o treinamento, além de uma memória de curto prazo manifestada pelas ativações transitórias transmitidas entre os nós. As LSTMs, por sua vez, expandem essa arquitetura ao incorporar uma célula de memória que atua como um núcleo de armazenamento intermediário. Essa célula é constituída por unidades interconectadas que utilizam mecanismos multiplicativos para regular, de maneira precisa, quais informações devem ser propagadas para os estados subsequentes ou descartadas. Nesse sentido, a Equação 7 define o cálculo necessário para a manutenção desse estado em um determinado instante t.

$$h_t = f\left(Ux_t + Wh_{t-1}\right) \tag{7}$$

onde h_t denota o estado oculto no instante de tempo t, elemento que representa a memória da RNN ao incorporar tanto a informação da entrada atual quanto o contexto acumulado de instantes anteriores. O termo x_t corresponde ao vetor de entrada, cuja projeção para o espaço do estado oculto é mediada pela matriz de pesos U. De forma simultânea, h_{t-1} refere-se ao

estado oculto do instante anterior, o qual preserva o histórico da sequência e é processado pela matriz de pesos recorrentes W. Por fim, f designa a função de ativação responsável por introduzir não linearidade ao modelo.

A arquitetura BiLSTM é obtida por meio da implementação de duas camadas de LSTM unidirecionais que operam sobre a mesma entrada em sentidos opostos. Dada uma sequência de *tokens* definida por x $= (x_1, x_2, ..., x_n)$, a primeira camada processa a entrada na ordem original, de x_1 a x_n, ao passo que a segunda camada realiza o processamento de forma invertida, partindo de x_n até x_1. A representação final da BiLSTM é gerada pela concatenação das saídas correspondentes de ambas as camadas para cada passo temporal, permitindo que o modelo capture informações contextuais tanto precedentes quanto subsequentes a cada elemento da sequência.

A arquitetura BiLSTM adotada neste trabalho fundamenta-se em uma rede recorrente bidirecional de camada única, integrada a um classificador não linear. De forma análoga à estrutura da CNN detalhada anteriormente, o modelo inicia-se com uma camada de *embeddings* que mapeia cada *token* do vocabulário para um vetor denso de 256 dimensões, utilizando o índice zero para o *padding*. Essas representações alimentam uma camada LSTM bidirecional configurada com 384 unidades ocultas em cada sentido, o que possibilita a captura de dependências contextuais tanto na ordem direta quanto na inversa da sequência. Ao final do processamento recorrente, os estados ocultos correspondentes ao último passo temporal de ambas as direções são concatenados, resultando em um vetor de características de 768 dimensões. Esta representação é então submetida ao classificador FNN, o qual aplica uma camada de *dropout* para regularização e uma projeção linear para o espaço de classes, finalizada pela função de ativação GELU, cujos *logits* resultantes subsidiam a etapa de classificação.

3.3.1.3 Bidirectional Encoder Representations from Transformers

Diferentemente das BiLSTMs, que concatenam camadas unidirecionais opostas, o BERT baseia-se nos codificadores da arquitetura Transformer para o pré-treinamento de representações bidirecionais. Esse modelo é projetado para condicionar a representação de cada *token* simultaneamente aos seus contextos à esquerda e à direita, superando as limitações de abordagens predecessoras que empregavam modelos estritamente unidirecionais ou a combinação superficial destes. Para viabilizar essa bidirecionalidade profunda, o modelo utiliza o objetivo de modelagem de linguagem mascarada, ou Masked Language Modeling (MLM) em inglês, uma técnica que se assemelha a uma tarefa de eliminação de ruído.

A MLM constitui a tarefa central projetada para viabilizar o treinamento de modelos inerentemente bidirecionais, como o BERT. Nesse procedimento, as sequências de entrada são processadas de modo que uma fração de seus *tokens* é substituída pelo *token* especial [MASK], incumbindo ao modelo a predição desses elementos ocultos a partir do contexto bidirecional fornecido pelos demais componentes da sequência (CASELI; NUNES, 2024). Durante essa tarefa,

o modelo pode inferir termos que, embora divirjam dos originais, permanecem semanticamente plausíveis dentro do contexto. Essa característica, conforme ressaltado por Caseli e Nunes (2024), torna a avaliação de modelos de linguagem baseados em predição de texto um desafio complexo, dada a subjetividade e a multiplicidade de saídas válidas para uma mesma lacuna contextual.

Concluída a fase de pré-treinamento, o modelo pode ser adaptado para a tarefa final de interesse por meio de um processo de ajuste fino, o qual demanda modificações mínimas em sua arquitetura original. Nesse estágio, as representações geradas pelo BERT são encaminhadas a uma camada de saída adicional, configurada de acordo com a natureza da aplicação pretendida. Subsequentemente, todos os parâmetros do codificador dos Transformers pré-treinado são refinados, ao passo que a camada de saída suplementar é treinada integralmente a partir de parâmetros iniciais, permitindo que o modelo se especialize nos requisitos específicos do problema abordado.

Estruturalmente, os codificadores dos Transformers são compostos por l camadas idênticas, conforme a arquitetura proposta por Vaswani et al. (2017) para superar as limitações dos modelos recorrentes tradicionais, como as LSTMs. A principal motivação para o desenvolvimento dos Transformers reside na viabilização de uma paralelização mais eficiente, substituindo a natureza intrinsecamente sequencial das RNNs por mecanismos exclusivos de atenção (SOUZA; NOGUEIRA; LOTUFO, 2020b). Enquanto o caráter sequencial das abordagens anteriores restringe o processamento paralelo dentro de um mesmo exemplo de treinamento e impõe desafios de memória em sequências extensas, os codificadores operam mapeando uma sequência de *tokens* de entrada x $= (x_1, x_2, ..., x_n)$ em uma sequência de representações vetoriais z $= (z_1, z_2, ..., z_n)$.

Cada camada do codificador subdivide-se em dois componentes principais: uma subcamada de *multi-head self-attention* e uma FNN. No mecanismo de self-attention, as saídas da camada precedente são processadas como consultas, ou *queries* em inglês, chaves, ou *keys* em inglês, e valores, ou *values* em inglês, permitindo o cálculo de uma soma ponderada dos valores a partir da similaridade entre consultas e chaves. Essa configuração assegura que cada posição no codificador possa estabelecer relações de dependência com todas as posições da camada anterior simultaneamente.

A arquitetura dos modelos baseados em BERT adotada neste trabalho segue a configuração base do codificador do Transformer, sendo utilizados o BERTimbau e o mBERT. Ambas as variantes compartilham uma estrutura comum, composta por uma camada inicial de *embeddings* seguida pelo empilhamento de 12 camadas de codificadores. Nessa arquitetura, a camada de *embeddings* sintetiza informações de *tokens*, posições e segmentos em vetores densos de dimensão 768, que alimentam os blocos subsequentes.

Cada camada do codificador integra um mecanismo de *multi-head self-attention* de 12 *heads*, responsável pela modelagem de dependências contextuais globais, seguido de uma

FNN totalmente conectada, ambos operando com conexões residuais e normalização de camada. Embora os estados ocultos mantenham a dimensionalidade de 768 em toda a rede e os codificadores sejam arquiteturalmente equivalentes, a distinção central entre os modelos repousa na camada de *embeddings*. Enquanto o BERTimbau é especializado no idioma português, o mBERT apresenta um vocabulário e um volume de parâmetros superiores, visto que sua camada de entrada é compartilhada entre aproximadamente 104 línguas distintas.

3.3.1.4 DistilBERT

O DistilBERT constitui uma arquitetura baseada nos codificadores dos Transformers e otimizada por meio da KD do BERT, visando a obtenção de um modelo mais compacto, rápido e eficiente. Esta abordagem resulta em uma rede com 40% menos parâmetros e 60% mais rápido que a versão original, mantendo, simultaneamente, mais de 95% da eficácia preditiva do modelo professor. Estruturalmente, o DistilBERT utiliza apenas metade das camadas do BERT, com uma estratégia de inicialização de pesos que consiste na seleção intercalada de uma a cada duas camadas do modelo de referência, assegurando a preservação da capacidade de representação durante a compressão.

A arquitetura dos modelos baseados em DistilBERT adotada neste trabalho segue o paradigma dos modelos BERT precedentes, sendo empregadas as variantes DistilBERT e DistilBERT Multilíngue (DistilmBERT). Ambas as estruturas fundamentam-se no codificador do Transformer e preservam a configuração dimensional das versões base, porém com uma estrutura mais compacta: diferentemente do BERT original, que utiliza 12 camadas de codificadores, o DistilBERT é composto por apenas seis camadas, reduzindo a profundidade da rede pela metade para otimizar a eficiência computacional. Inicialmente, uma camada de *embeddings* combina informações de *tokens* e posições em vetores densos de dimensão 768, que alimentam os blocos de codificação.

Estes modelos são constituídos por seis camadas empilhadas de codificadores, nas quais cada camada integra um mecanismo de *multi-head self-attention* de 12 *heads* e uma FNN totalmente conectada, ambas acompanhadas por conexões residuais e normalização de camada. Ao longo de toda a arquitetura, os estados ocultos preservam a dimensionalidade de 768. A diferença central entre as variantes reside na camada de *embeddings*: no DistilmBERT, essa camada é consideravelmente mais extensa, visto que o modelo resulta da destilação do mBERT e mantém a capacidade de representação multilíngue via vocabulário ampliado, ao passo que o DistilBERT convencional foca em um vocabulário monolíngue mais delimitado.

3.3.1.5 Mixture of Experts

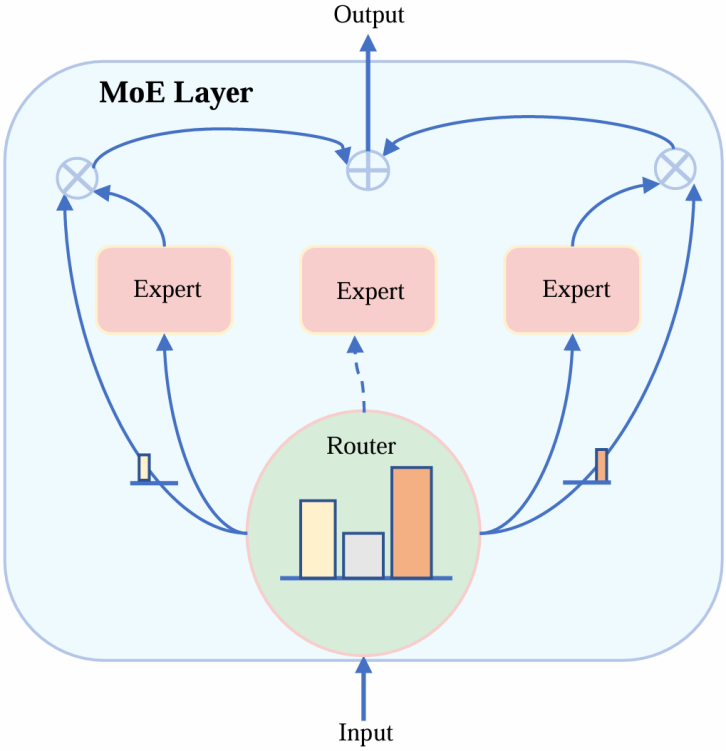
A arquitetura MoE fundamenta-se na estratégia de "dividir para conquistar" e representa uma mudança de paradigma em relação aos modelos densos convencionais, caracterizados pela ativação de todos os seus parâmetros para cada entrada processada. Sob uma perspectiva

estrutural, o sistema define-se pela integração de três componentes centrais, sendo eles uma função de roteamento, ou *gating function* em inglês, um conjunto de redes especialistas e um mecanismo de decisão responsável pela interação entre essas partes. A *gating function* atua como um coordenador matemático ao mapear os dados de entrada para os especialistas mais pertinentes, processo que frequentemente emprega funções lineares combinadas com operações de *softmax* e a seleção dos k elementos de maior pontuação, técnica conhecida como *top-k*. Essa seleção esparsa permite que o modelo direcione o fluxo de informação exclusivamente para sub-redes especializadas em domínios de conhecimento particulares, o que favorece o desenvolvimento de competências modulares e previne a interferência entre conhecimentos heterogêneos.

Na prática de redes neurais profundas, a exemplo dos modelos baseados em Transformers, o mecanismo MoE é majoritariamente implementado mediante a substituição das camadas FNN por camadas MoE. Tal escolha técnica justifica-se pela maior esparsidade e especificidade de tarefas observadas nessas estruturas quando comparadas às camadas de self-atention. Segundo Mu e Lin (2025), esse arcabouço viabiliza a expansão massiva da capacidade do modelo e permite alcançar escalas de trilhões de parâmetros sem acarretar um aumento proporcional na carga computacional, estabelecendo um equilíbrio otimizado entre desempenho preditivo e eficiência sistêmica.

A Figura 7 ilustra uma esquemática simples de uma arquitetura MoE padrão.

Figura 7 – Representação esquemática de uma arquitetura Mixture of Experts.



Fonte: Mu e Lin (2025).

A *gating function* constitui o núcleo matemático do mecanismo de roteamento em arquiteturas MoE e é responsável pela alocação estratégica dos dados de entrada aos especialistas mais pertinentes. No âmbito do desenvolvimento dessas funções, a literatura define como critérios fundamentais a capacidade de discernir com precisão as características tanto dos dados quanto dos especialistas, bem como a promoção de uma distribuição de carga equitativa. Essa medida busca mitigar o fenômeno do colapso do modelo, cenário no qual a subutilização de determinados componentes compromete a separação de conhecimentos heterogêneos (MU; LIN, 2025). Em termos convencionais, adota-se predominantemente o roteamento linear com ativação *softmax*, cuja formulação é dada por $G(x)i = \text{softmax}(\text{top-k}(g(x)+\mathcal{R}\text{noise}, k))i$, onde o componente de ruído $\mathcal{R}$noise é inserido para estimular a exploração de distintos especialistas e prevenir a dependência excessiva de subconjuntos específicos.

A eficácia dessa operação, segundo Mu e Lin (2025), demonstra sensibilidade à ordem de execução dos componentes. A aplicação do operador *top-k* previamente ao *softmax* favorece a eficiência computacional ao filtrar especialistas irrelevantes, ao passo que a inversão dessa sequência proporciona pesos de ativação com maior rigor estatístico na distribuição de probabilidade. Adicionalmente às abordagens lineares, emergem paradigmas não lineares direcionados a desafios específicos, a exemplo do roteamento baseado em similaridade de cosseno para tarefas de generalização de domínio e do conceito de Soft MoE. Este último substitui a alocação discreta de *tokens* por médias ponderadas com o intuito de contornar dificuldades de otimização e evitar o descarte inadvertido de dados.

Tais refinamentos teóricos na função de roteamento mostram-se determinantes para viabilizar o escalonamento da capacidade do modelo, pois permitem que a modularidade e a especialização resultem em ganhos de desempenho sem acarretar um incremento proporcional no custo computacional de treinamento e inferência. As redes especialistas atuam como subconjuntos de parâmetros que se especializam em domínios de conhecimento distintos por meio da alocação dinâmica de dados de entrada. Apesar da possibilidade de operarem como modelos independentes, a prática atual prioriza a integração desses componentes em arquiteturas neurais consolidadas, substituindo camadas específicas para otimizar o desempenho e a eficiência computacional. Formalmente, uma camada MoE é definida pela expressão $MoE(x) = \sum_{i \in ID} w_i M_i(x)$, onde ID identifica os especialistas selecionados pela função de roteamento, w_i corresponde ao peso de ativação atribuído e $M_i(x)$ denota o modelo do especialista. Este último é frequentemente implementado sob a forma de uma FNN de duas camadas.

A predominância da substituição da camada FNN em arquiteturas Transformer por blocos MoE fundamenta-se na esparsidade intrínseca dessas estruturas e no fenômeno de modularidade emergente, no qual ativações neuronais específicas demonstram correlação direta com tarefas particulares. Tal característica permite que a estrutura MoE reflita a natureza modular e a especificidade de domínio do modelo. Expansões dessa abordagem

englobam a Mixture-of-Attention, que emprega *attention heads* como especialistas para reduzir a redundância computacional e escalar o número de parâmetros, bem como o uso de especialistas convolucionais. Estes aproveitam a correlação espacial e a estrutura hierárquica de dados visuais para decompor problemas de classificação complexos em subproblemas de granularidade fina.

O mecanismo de roteamento constitui o componente decisório fundamental que define a granularidade e a frequência com que os dados de entrada são direcionados a sub-redes especializadas, sendo essencial para adequar o modelo às exigências de tarefas específicas. Conforme a taxonomia apresentada, o roteamento pode ser operado em diversos níveis: (i) o nível de *token* representa a abordagem clássica, na qual decisões individuais são tomadas para unidades discretas de texto ou *patches* de imagem, visando preservar a diversidade das representações e capturar nuances contextuais; (ii) o nível de modalidade organiza o fluxo de dados com base na natureza sensorial da entrada, emulando codificadores especializados para minimizar a interferência entre domínios heterogêneos; e (iii) o nível de tarefa utiliza identificadores específicos para isolar o processamento de diferentes objetivos, o que otimiza significativamente o uso de memória e os custos de comunicação durante a inferência ao carregar apenas os especialistas pertinentes à tarefa em execução.

Além dessas, estratégias de roteamento em nível de contexto empregam mecanismos de *global pooling* para informar o roteador sobre o ambiente semântico circundante, enquanto o roteamento por atributos incorpora embutimentos binários multidimensionais, abrangendo variáveis como tipo de causalidade e fonte do dado, para viabilizar o tratamento de estruturas informacionais complexas. Essa flexibilidade no *design* do roteamento sustenta abordagens metodológicas avançadas, permitindo que o modelo equilibre a especialização dos especialistas com a eficiência computacional necessária para o processamento de grandes volumes de dados.

A arquitetura baseada em MoE proposta neste trabalho integra uma camada de *embeddings*, um mecanismo de roteamento e um conjunto de especialistas, onde cada unidade é estruturada como um classificador BiLSTM não linear análogo ao descrito na subseção 3.3.1.2. Em conformidade com as arquiteturas detalhadas anteriormente, a camada de *embeddings* mapeia os *tokens* em vetores densos de 256 dimensões e reserva o índice zero para a representação do *padding*. Para a operação do *gate*, calcula-se a representação média da sequência na dimensão temporal, a qual serve de entrada para uma camada FNN que projeta o resultado em um vetor de pontuações associado aos especialistas disponíveis. A partir desses *scores*, selecionam-se os k especialistas mais relevantes, cujos valores são normalizados via função *softmax* para a obtenção de pesos probabilísticos.

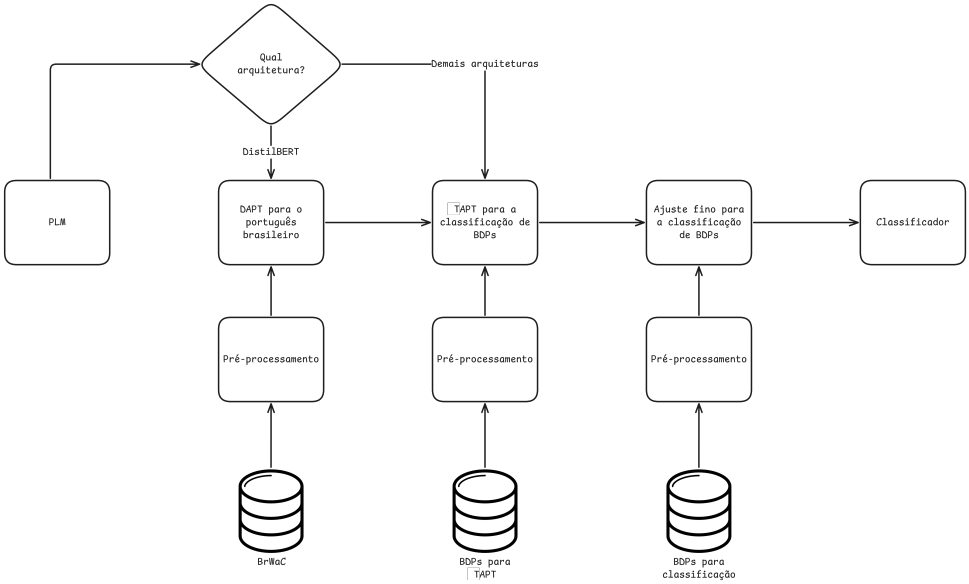
Cada unidade selecionada processa a sequência de maneira independente e, internamente, consiste em uma BiLSTM de camada única composta por 384 unidades ocultas por direção. Os estados ocultos finais dessa etapa são concatenados em um vetor de 768 dimensões e submetidos a uma camada FNN com *dropout* e ativação GELU. Por fim, as saídas dos especialistas são combinadas mediante uma soma ponderada pelos pesos do roteamento, o que configura um

mecanismo de roteamento em nível de contexto e gera os *logits* finais para a classificação.

3.3.2 Treinamento dos modelos

A etapa de treinamento dos modelos é condicionada pela arquitetura neural selecionada para a classificação dos BDPs e organiza-se em três *pipelines* distintos. O primeiro desses fluxos fundamenta-se na utilização de PLMs como ponto de partida, conforme ilustrado na Figura 8, e abrange todas as variações baseadas nas arquiteturas BERT e DistilBERT previamente discutidas.

Figura 8 – Fluxo de treinamento continuado de Pre-trained Language Models.



Fonte: Elaborada pelo autor.

Inicialmente, a definição dos pesos da rede ocorre com base em um dos PLMs, os quais são treinados em vastos *corpora* não rotulados de domínio geral. Caso a arquitetura selecionada seja o DistilBERT, realiza-se uma etapa de adaptação via DAPT previamente ao TAPT destinado à classificação dos BDPs, com o objetivo de ajustar o modelo ao contexto da língua portuguesa brasileira. Em contrapartida, as demais arquiteturas dispensam esse passo intermediário, haja vista que já foram expostas a dados em português durante o pré-treinamento original. Para a referida fase de adaptação, utiliza-se o conjunto de dados brWaC sob a tarefa de MLM, empregando-se a função de perda de CE conforme preconizado por Souza, Nogueira e Lotufo (2020a) e expressa na Equação 8.

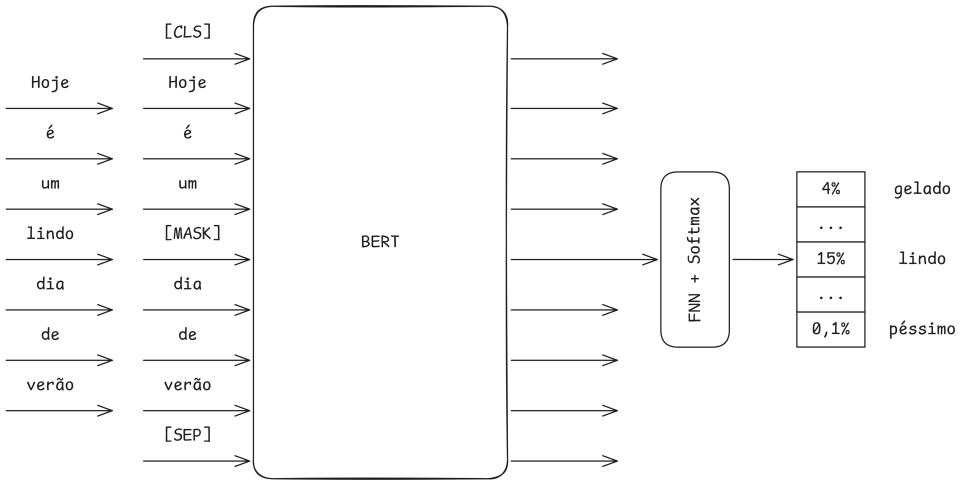
$$\mathcal{L}_{\text{CE}} = -\frac{1}{N}\sum_{i=1}^{N}(y_i \log \hat{y}_i), \tag{8}$$

onde N denota o número total de amostras no conjunto de dados, y refere-se ao vetor *one-hot* correspondente ao rótulo da amostra e $\hat{y}$ indica o vetor da distribuição de probabilidades

estimada pelo modelo.

A Figura 9 ilustra o fluxo de processamento da MLM. O processo tem início com a etapa de tokenização da sequência de entrada, exemplificada na figura pela frase "Hoje é um lindo dia de verão". A essa sequência, são acrescidos *tokens* especiais que auxiliam na estruturação computacional do texto, a saber, o *token* [CLS], posicionado no início para agregar a representação semântica global da entrada, e o *token* [SEP], responsável por demarcar o término do segmento. A essência do objetivo MLM consiste em corromper intencionalmente a entrada ao ocultar de forma estocástica uma proporção dos *tokens* da sequência original, taxa tradicionalmente fixada em 15%. O intuito dessa estratégia é compelir o modelo a inferir o elemento ausente com base exclusivamente no contexto circundante. No exemplo apresentado, o *token* correspondente à palavra "lindo"é selecionado e substituído pelo *token* especial [MASK].

Figura 9 – Fluxo de treinamento continuado de Pre-trained Language Models.



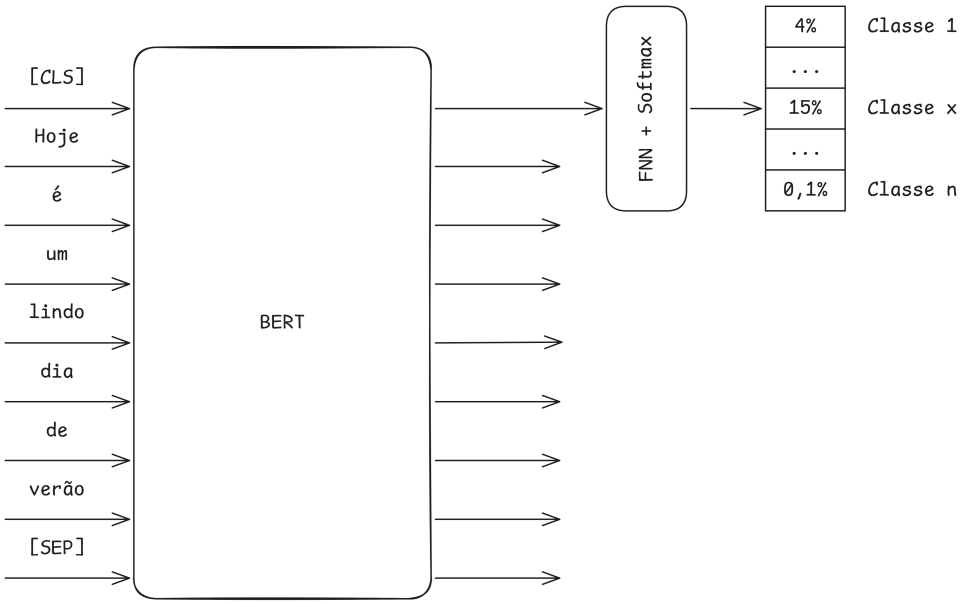
Fonte: Elaborada pelo autor.

A sequência modificada é, então, submetida às camadas de codificação do BERT. O mecanismo de *self-attention* bidirecional permite que a rede compute as representações vetoriais considerando, de forma simultânea, os contextos à esquerda e à direita do *token* mascarado. Para a predição final, o vetor de saída correspondente especificamente à posição do *token* [MASK] é projetado para a dimensão do vocabulário por meio de uma FNN. Em seguida, aplica-se a função de ativação Softmax, que converte os *logits* em uma distribuição de probabilidade sobre todas as palavras do vocabulário conhecido. Essa saída indica a probabilidade de cada palavra ser a candidata correta para preencher a lacuna. Conforme observado na representação, o modelo atribui probabilidades a diversos termos, a exemplo de 4% para a palavra "gelado"e 0,1% para "péssimo", ao passo que confere a maior probabilidade, correspondente a 15%, à palavra original "lindo". Durante o treinamento, a função de perda penaliza predições incorretas e ajusta os pesos da rede por meio de *backpropagation*, de modo que o modelo aprenda as representações linguísticas latentes.

No que concerne às demais arquiteturas, prescinde-se da etapa inicial de DAPT e avança-se diretamente para o TAPT, aplicado a um extenso *corpus* de textos não rotulados específicos do contexto de classificação de BDPs. Essa estratégia visa aprofundar o conhecimento especializado do modelo acerca da tarefa e favorecer o incremento do desempenho preditivo. Durante essa etapa, o treinamento permanece focado na tarefa de MLM, embora utilize as descrições de BDPs não rotuladas como base para o refinamento. Subsequentemente, a fase de ajuste fino adapta o modelo aos dados supervisionados ao empregar a estrutura e os parâmetros previamente calibrados pelo TAPT. Nesse estágio, o ajuste ocorre de ponta a ponta e explora a capacidade de generalização das redes neurais profundas para adequar os pesos à tarefa de classificação final mediante a otimização da função de CE.

A Figura 10 ilustra o fluxo de processamento para a tarefa de classificação de sentenças por meio do modelo BERT. Em contraste com a etapa de pré-treinamento por MLM, cujo propósito consiste na apreensão genérica da estrutura linguística, a referida arquitetura tem como objetivo direcionar o conhecimento latente do modelo para a predição de categorias específicas ao longo da fase de ajuste fino.

Figura 10 – Fluxo de treinamento continuado de Pre-trained Language Models.



Fonte: Elaborada pelo autor.

A entrada do modelo é constituída pela sequência de *tokens*, iniciada obrigatoriamente pelo *token* especial [CLS], indicativo da classificação. Durante o processamento por meio das múltiplas camadas de *self-attention* bidirecional, o vetor de estado oculto, correspondente ao *token* [CLS] atua como um agregador que sintetiza a representação semântica global de toda a sentença inserida. Para a predição da classe final, os vetores de saída correspondentes aos demais *tokens* do texto são desconsiderados. Nesse sentido, apenas o vetor final associado ao *token* [CLS] é extraído e projetado em uma FNN. A camada de saída emprega a função de

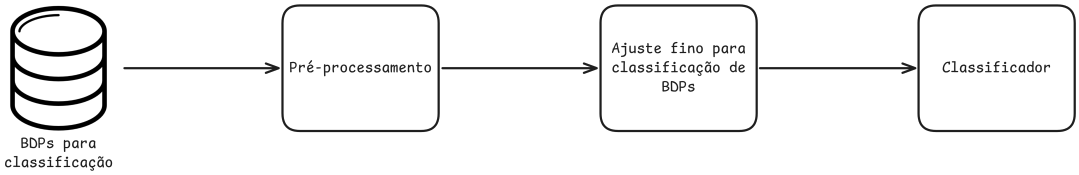
ativação Softmax para transformar os valores brutos em uma distribuição de probabilidade sobre o conjunto de classes. Por fim, o modelo classifica a sequência ao atribuí-la à categoria que apresentar a probabilidade predita mais elevada.

O pré-processamento aplicado aos conjuntos de dados em todos os *pipelines* descritos segue um protocolo padronizado, fundamentado na tokenização dos textos para viabilizar sua ingestão pelas redes neurais. Nesse contexto, as sequências que excedem o limite máximo de *tokens* estabelecido pela arquitetura são submetidas ao truncamento. Em contrapartida, aquelas com extensão inferior a esse limiar são complementadas com *tokens* de preenchimento, técnica denominada *padding*, a qual assegura a uniformidade dimensional das entradas indispensável para as etapas de treinamento e inferência dos modelos.

A tokenização consiste na segmentação do texto em unidades linguísticas mínimas, denominadas *tokens*, as quais integram um vocabulário preestabelecido. Embora esse conceito seja frequentemente associado à palavra completa, o processo pode ser aplicado em diferentes níveis de granularidade, variando desde caracteres isolados até subpalavras resultantes da fragmentação de termos em segmentos menores, como prefixos, sufixos ou sequências de caracteres de comprimento variável. O emprego de subpalavras tem por objetivo restringir o vocabulário do modelo a um tamanho finito e preservar, simultaneamente, a capacidade de representar um número ilimitado de *types*, conceito definido por Caseli e Nunes (2024) como os *tokens* distintos identificados em um *corpus*. Sob essa ótica, as palavras mais frequentes são codificadas integralmente, ao passo que os termos raros são decompostos e representados pela combinação de subpalavras preexistentes no vocabulário.

A segunda abordagem baseia-se no treinamento de modelos a partir do zero, conforme ilustrado no *pipeline* da Figura 11, e abrange as arquiteturas CNN, BiLSTM, MoE. Diferentemente da estratégia anterior, esse método emprega parâmetros inicializados aleatoriamente e dispensa o uso de aprendizado por transferência. Consequentemente, o processo concentra-se na etapa de ajuste fino, executada de modo análogo ao descrito no fluxo precedente, porém sem o benefício de pesos pré-ajustados em *corpora* de domínio geral.

Figura 11 – Fluxo de treinamento de modelos inicializados do zero.

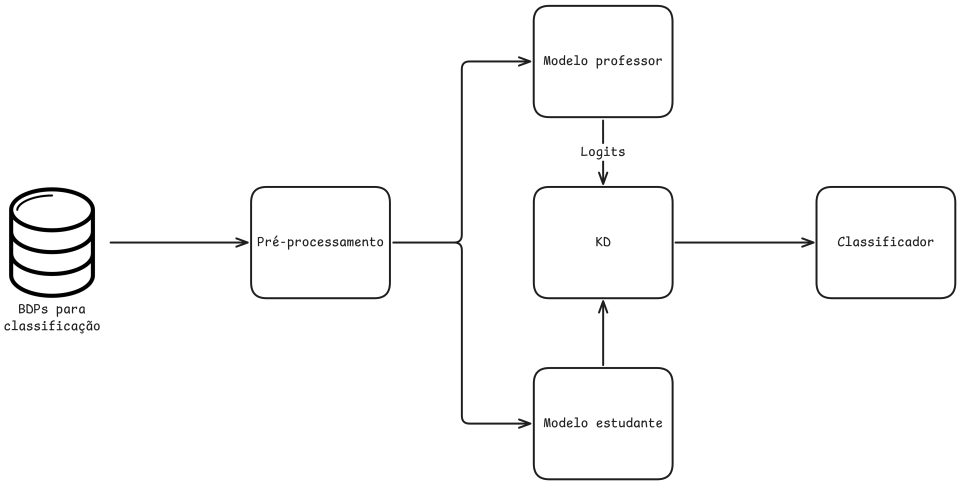


Fonte: Elaborada pelo autor.

Por fim, os modelos treinados via KD seguem o procedimento agnóstico em relação à arquitetura ilustrado na Figura 12. A distinção central dessa abordagem reside no aproveitamento de classificadores previamente treinados, segundo as estratégias descritas anteriormente, para

atuarem como professores durante a etapa de ajuste fino. Nesse processo, o modelo estudante, embora inicializado aleatoriamente, é submetido ao treinamento sob a supervisão do modelo professor, mecanismo que viabiliza a transferência de conhecimento e a captura de padrões complexos por uma estrutura computacionalmente mais eficiente.

Figura 12 – Fluxo de treinamento utilizando destilação de conhecimento.



Fonte: Elaborada pelo autor.

A técnica de KD promove a transferência de conhecimento no nível das saídas, ou *outputs* em inglês, ao orientar o modelo estudante a reproduzir o comportamento da rede professora diante das amostras de treinamento. Em consonância com a abordagem de Tang et al. (2019), utiliza-se um objetivo ponderado que integra a minimização da função de MSE, relativa à destilação, à minimização da CE, voltada à otimização da rede estudante, conforme detalhado na Equação 9.

$$\begin{aligned} \mathcal{L} &= \alpha \times \mathcal{L}_{\text{CE}} + (1-\alpha) \times \mathcal{L}_{\text{distill}}, \\ \mathcal{L}_{\text{distill}} &= \frac{1}{N}\sum_{i=1}^{N}(z_i^{(T)} - z_i^{(S)})^2, \end{aligned} \tag{9}$$

onde N indica o número de amostras do conjunto de dados. As variáveis $\mathcal{L}_{\text{CE}}$ e $\mathcal{L}_{\text{distill}}$ correspondem às funções de custo de CE e de destilação, ao passo que o coeficiente α regula a contribuição de cada termo na composição da perda final. Os vetores $z^{(T)}$ e $z^{(S)}$ designam os *logits* produzidos pelas redes professora e estudante, respectivamente.

3.4 Protocolo experimental

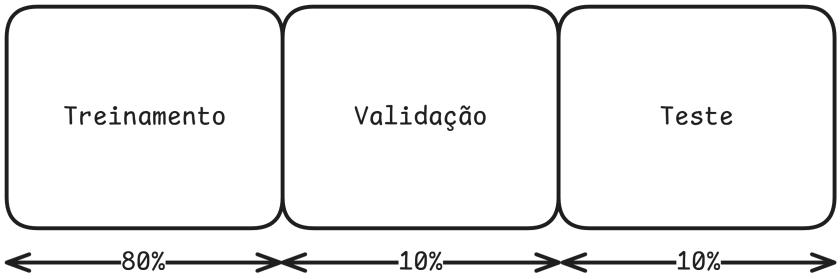
Esta seção descreve a infraestrutura computacional e o ambiente de software adotados para a condução dos experimentos, realizados com base na média de cinco execuções independentes. O ambiente de hardware consistiu em uma estação de trabalho com sistema

operacional Ubuntu 23.10, equipada com processador Intel Xeon Silver 4214R de 12 núcleos físicos e frequência de $2{,}40$ GHz, GPU NVIDIA RTX A4000 com 16 GB de memória e 128 GB de memória principal. O desenvolvimento do código foi realizado em Python $3.12.3$, integrando as bibliotecas numpy $1.26.4$ e datasets $2.19.2$ para a manipulação de *arrays* e dados, além do pacote ftfy $6.2.0$ para a limpeza textual. A implementação e o treinamento das arquiteturas de DL fundamentaram-se no uso de torch $2.3.1$ e transformers $4.41.2$, enquanto o cálculo das métricas de classificação e eficiência foi processado, respectivamente, pelas bibliotecas scikit-learn $1.5.0$, deepspeed $0.14.4$ e codecarbon $3.2.0$.

3.4.1 Divisão dos dados

Para a condução dos experimentos, o conjunto de BDPs foi segmentado em três subconjuntos, obedecendo à proporção de 80/10/10. Nessa configuração, 80% dos dados foram destinados ao treinamento, 10% à validação e os 10% remanescentes à etapa de teste, conforme detalha o esquema apresentado na Figura 13.

Figura 13 – Estratégia de divisão do conjunto de dados de Boletins Diários de Perfuração.



Fonte: Elaborada pelo autor.

Nesse arranjo, o conjunto de validação é empregado para monitorar o desempenho do modelo durante a otimização de hiperparâmetros, etapa detalhada na subseção 3.4.2, bem como para viabilizar a aplicação da técnica de parada antecipada, ou *early stopping* em inglês. Em contrapartida, o conjunto de teste é reservado exclusivamente para a avaliação final das arquiteturas, medida que assegura uma análise imparcial do desempenho e da capacidade de generalização dos classificadores obtidos.

3.4.2 Configurações de treinamento

Dadas as substanciais diferenças estruturais entre as arquiteturas analisadas, as quais impedem uma comparação direta estritamente equitativa, a definição dos hiperparâmetros não teve por objetivo a eliminação de discrepâncias intrínsecas, mas a garantia de consistência metodológica no processo de treinamento. Nesse sentido, adotaram-se configurações uniformes em todas as etapas, conforme detalhado na Tabela 1. Tais escolhas fundamentaram-se nos protocolos estabelecidos por Souza, Nogueira e Lotufo (2020a) para o DAPT no contexto do

português brasileiro e por Rodrigues et al. (2022) para as fases de TAPT e de ajuste fino na classificação de BDPs. Da mesma forma, ao considerar que a aplicação da KD representa um caso particular de ajuste fino cuja função objetivo pondera a relação entre predições, rótulos reais e *logits* do professor, as mesmas configurações experimentais foram preservadas para essa modalidade.

Tabela 1 – Configuração dos hiperparâmetros utilizados no treinamento dos modelos.

Hiperparâmetros	**Treinamento continuado**		**Ajuste fino**
	DAPT	**TAPT**	**e KD**
Batch size	16	8	12
Passos de acumulação de gradientes	8	–	–
Épocas	8	8	1.000
Warmup steps	10.000	–	–
Paciência para parada antecipada	–	–	5
Limiar de parada antecipada	–	–	1×10^{-4}
Algoritmo de otimização	AdamW	AdamW	AdamW
Decaimento de peso	0.01	0.01	0.01
Taxa de aprendizado	1×10^{-4}	5×10^{-5}	4×10^{-5}
Escalonador de taxa de aprendizado	Linear	Linear	Linear
Comprimento máximo das sequências	512	256	256

Fonte: Elaborada pelo autor.

O tamanho do lote, ou *batch size* em inglês, define o número de amostras processadas antes de cada atualização nos parâmetros do modelo. Especificamente para o processo de DAPT, adotou-se um *batch size* de 16 amostras, valor que foi reduzido para 8 durante o TAPT. Nas etapas subsequentes, referentes ao ajuste fino e à KD, essa configuração foi fixada em 12 amostras.

O número de épocas define a quantidade total de passagens completas pelo conjunto de dados. Para as etapas de DAPT e TAPT, estabeleceu-se o valor de 8 épocas. Nas fases de ajuste fino e de KD, esse limite foi ampliado para 1.000 com o intuito de possibilitar a convergência do modelo, a qual é regulada por mecanismos de parada antecipada. Adicionalmente, os passos de acumulação de gradientes determinam o intervalo de iterações em que os gradientes são retidos antes da atualização efetiva dos parâmetros. Nessa configuração, o DAPT utilizou 8 passos de acumulação, ao passo que esse recurso não foi empregado no TAPT, no ajuste fino nem na KD.

A paciência para parada antecipada determina o limite de épocas consecutivas sem melhoria na métrica de validação e atua como critério de interrupção do treinamento para mitigar o sobreajuste. Nas etapas de ajuste fino e de KD, esse parâmetro foi fixado em 5 épocas, ao passo que nos estágios de DAPT e TAPT ele não é empregado. De maneira complementar, o limiar de parada antecipada estabelece a variação mínima necessária no desempenho de validação para que se considere a existência de progresso. Para o ajuste fino e a KD, definiu-se esse valor em 1×10^{-4}. Sob essa configuração, o contador de épocas estagnadas é incrementado

sempre que o ganho observado for inferior ao limiar preestabelecido, garantindo a finalização eficiente do processo quando a rede alcança a estabilidade.

A taxa de aprendizado estabelece a magnitude dos ajustes aplicados aos parâmetros da rede durante o processo de otimização. Para o estágio de DAPT, adotou-se o valor de 1×10^{-4}, ao passo que para o TAPT e a KD as taxas foram fixadas em 5×10^{-5} e 4×10^{-5}, respectivamente. A variação desse parâmetro ao longo do tempo é controlada por um escalonador linear, o qual promove o decaimento gradual e constante da taxa após o período inicial. Adicionalmente, a etapa de DAPT contempla uma fase de aquecimento, ou denominada *warmup steps* em inglês, composta por 10.000 passos, na qual a taxa de aprendizado é incrementada progressivamente. Essa estratégia de inicialização, contudo, não é aplicada nas fases de TAPT, ajuste fino e KD.

No que concerne à otimização dos parâmetros, adotou-se o algoritmo AdamW (LOSHCHILOV; HUTTER, 2019) em todas as abordagens de treinamento. Esse método estocástico distingue-se da implementação convencional do Adam ao promover o desacoplamento entre o decaimento de peso e as atualizações baseadas em gradientes, característica que permite mitigar limitações de convergência observadas na versão original. O decaimento de peso atua como uma técnica de regularização ao introduzir um termo de penalidade na função de perda com o intuito de prevenir o sobreajuste e, para o conjunto de experimentos realizados, esse hiperparâmetro foi fixado em $0{,}01$.

Por fim, o comprimento máximo das sequências estabelece o limite superior para a extensão das entradas processadas pelas redes neurais. Para a etapa de DAPT, definiu-se o limite em 512 *tokens*, enquanto nas fases de TAPT e de KD esse parâmetro foi reduzido para 256 *tokens*. Essa adaptação visa compatibilizar o processamento com a extensão característica das descrições de BDPs e otimizar tanto o consumo de recursos computacionais quanto a eficiência do treinamento, sem que haja comprometimento da integridade das informações extraídas.

3.4.3 Procedimento de avaliação e validação

A estratégia de validação deste estudo estrutura-se sob uma abordagem multidimensional que transcende a análise convencional de desempenho preditivo ao incorporar métricas de eficiência computacional, impacto ambiental e rigor estatístico. A avaliação é conduzida nos três níveis hierárquicos de granularidade dos BDPs, especificamente *Atividade*, *Operação* e *Etapa*. Nesse contexto, os classificadores resultantes de cada *pipeline* de treinamento são submetidos à análise conforme as métricas apresentadas na seção 3.2. Adicionalmente, selecionou-se o modelo fundamentado na arquitetura BERT $\text{PetroBERT}_{\text{BERTimbau}}$ para atuar como rede professora no estágio de KD e como *baseline* comparativo, haja vista que essa arquitetura apresentou o melhor desempenho na tarefa de classificação no estudo conduzido por Rodrigues et al. (2022).

Para mensurar a eficácia dos modelos na classificação textual, adotou-se o F1-Score

como métrica principal. A opção por essa média justifica-se pela necessidade de tratar todas as classes com igual importância, independentemente do desbalanceamento inerente aos dados industriais. Complementarmente, reportam-se a precisão e a revocação com o objetivo de detalhar o comportamento dos classificadores em relação aos FPs e FNs. O desempenho apresentado corresponde à média aritmética obtida a partir de 5 execuções independentes, de modo a assegurar a generalização dos resultados.

Em consonância com os princípios de *Green AI*, o estudo quantificou o custo operacional das arquiteturas. As métricas coletadas para essa análise incluem:

- **Tempo de Inferência (s):** Corresponde ao tempo médio necessário para o processamento do conjunto de teste, indicador essencial para validar a viabilidade de execução em tempo real.
- **Consumo Energético (kWh) e Pegada de Carbono (gCO_2eq):** Estimam o impacto ambiental associado à fase de inferência e permitem o cálculo do compromisso, ou *trade-off*, entre precisão e sustentabilidade.
- **Complexidade do Modelo:** Mensurada a partir do número de parâmetros treináveis e da quantidade de operações de ponto flutuante, ou FLOPs.

A análise desses fatores culminou na construção de Fronteiras de Pareto. Segundo Deb (2011), no contexto da otimização multiobjetivo, essa fronteira representa o conjunto de soluções ótimas de compromisso, ou *trade-off* em inglês. Tal configuração emerge em cenários caracterizados por objetivos conflitantes, nos quais a melhoria de um aspecto implica necessariamente a degradação de outro. Matematicamente, a fronteira é composta por pontos que satisfazem a condição de não dominância, a qual estabelece que tais soluções não podem ser superadas por nenhuma outra em todos os objetivos simultaneamente (DEB, 2011). Essa delimitação permitiu a identificação, tanto visual quanto analítica, das arquiteturas que apresentam a melhor relação custo-benefício, tecnicamente categorizadas como soluções não dominadas.

Ademais, com o objetivo de aferir a confiabilidade dos modelos em um cenário de produção crítico, analisou-se a dispersão dos resultados obtidos durante a etapa de validação. Nesse contexto, a estabilidade foi quantificada por meio do Coeficiente de Variação (CV), métrica que expressa a variabilidade relativa dos dados. O cálculo desse coeficiente é definido pela razão conforme estabelecido na Equação 10:

$$CV = \left(\frac{\sigma}{\mu}\right) \times 100, \tag{10}$$

onde σ representa o desvio padrão dos resultados obtidos nas execuções e μ corresponde à média aritmética do F1-Score. O fator multiplicativo 100 é aplicado para expressar a variabilidade

relativa em uma escala percentual, o que facilita a análise da estabilidade entre as diferentes arquiteturas.

Com o intuito de garantir que as discrepâncias de desempenho observadas não sejam decorrentes de fatores aleatórios, os resultados foram submetidos a um rigoroso tratamento estatístico:

1. **Verificação de Normalidade:** A premissa de que os dados seguem uma distribuição Gaussiana é fundamental para a validade dos testes paramétricos. Para verificar essa condição, analisou-se a distribuição das amostras de F1-Score por meio do teste de *Shapiro-Wilk* (SHAPIRO; WILK, 1965). Este método avalia a hipótese nula de que a população é normalmente distribuída. Uma vez confirmada a normalidade majoritária dos dados, indicada por valores de $p > 0{,}05$, legitimou-se a opção pela utilização de testes paramétricos, os quais oferecem maior poder estatístico nessas condições.

2. **Teste de Hipótese:** Para comparar o desempenho de cada arquitetura proposta em relação ao *baseline* PetroBERT$_{\text{BERTimbau}}$, empregou-se o teste-t pareado com um nível de significância de $0{,}05$. Conforme a teoria estatística, este teste verifica a hipótese nula de que as médias das populações são iguais, expressa por $H_0 : m_A - m_B = 0$, assumindo-se homogeneidade de variância e normalidade (COHEN, 1988). A escolha pela versão pareada deve-se à necessidade de controlar a dependência entre as amostras, visto que os modelos foram avaliados nas mesmas partições de dados durante as execuções independentes, o que isola a variabilidade causada pelos *folds* da variabilidade intrínseca dos modelos.

3. **Correção de Múltiplas Comparações:** A realização simultânea de múltiplos testes de hipótese aumenta a probabilidade de cometer erros do Tipo I, ou falsos positivos, o que infla a Taxa de Erro Familiar, denotada pela sigla FWER. Para mitigar esse risco sem incorrer no conservadorismo excessivo da correção de Bonferroni padrão, aplicou-se o método sequencial de Holm-Bonferroni (HOLM, 1979). Este procedimento ajusta os valores-p de forma iterativa, ordenando-os do menor para o maior e aplicando critérios de rejeição progressivamente menos restritivos, garantindo assim o controle rigoroso da significância global da análise.

4. **Tamanho de Efeito:** Enquanto o p-valor indica se há uma diferença estatisticamente significativa, tal métrica não informa a magnitude ou a importância prática dessa divergência. Para suprir essa lacuna, calculou-se o d de Cohen. Teoricamente, esta métrica padroniza o tamanho do efeito ao dividir a diferença bruta entre as médias pelo desvio padrão comum das populações, resultando em uma grandeza adimensional livre da unidade de medida original (COHEN, 1988). Isso permite expressar a distância entre os desempenhos em unidades de variabilidade. Para a interpretação, adotaram-se os limiares convencionais propostos por Cohen (1988), nos quais $|d| \approx 0{,}2$ indica um efeito pequeno

e difícil de distinguir do ruído, $|d| \approx 0{,}5$ aponta um efeito médio visível a olho nu, e $|d| \geq 0{,}8$ denota um efeito grande, com separação clara entre as médias.

Por fim, o impacto da técnica de compressão foi isolado mediante o cálculo do $\Delta F1$, indicador que expressa a diferença absoluta de desempenho. Esse procedimento comparou diretamente as versões destiladas via KD com seus respectivos modelos base, com o objetivo de validar a eficácia da transferência de conhecimento.

4 Resultados e Discussão

Este capítulo apresenta uma análise multidimensional dos experimentos realizados ao estabelecer um comparativo entre as arquiteturas leves propostas e o modelo PetroBERT, descrito por Rodrigues et al. (2022). Esta arquitetura é adotada no presente estudo como *baseline*, justificando-se por ser a solução efetivamente implementada no âmbito do projeto ProtoRadiar, na Petrobras, e por representar o estado da arte industrial atualmente empregado na classificação automática dos BDPs. O objetivo central consiste em avaliar se as técnicas de compressão de modelos podem conciliar a alta capacidade de generalização inerente à arquitetura BERT com os rigorosos requisitos de eficiência operacional.

Com o objetivo de assegurar a validade e a reprodutibilidade das comparações, estabeleceram-se hiperparâmetros experimentais específicos. No treinamento via KD, atribuiu-se o valor de $0{,}2$ ao hiperparâmetro α, configuração que pondera as funções de custo de CE e de destilação em 20% e 80%, respectivamente. Tal decisão fundamentou-se nas diretrizes de Tang et al. (2019), haja vista que seus estudos indicam que a atribuição de maior ênfase ao objetivo da KD tende a resultar em um desempenho superior para os modelos estudantes. No tocante à precisão das estimativas, adotou-se um intervalo de confiança de 95%.

As avaliações de eficiência computacional foram conduzidas em ambiente controlado, configurando-se o processamento em lotes, ou *batches* em inglês, de 512 amostras para os testes de tempo de inferência e cálculo de FLOPs. Com o objetivo de mitigar variações pontuais e garantir a consistência estatística, registraram-se todas as métricas a partir de 5 execuções independentes, nas quais se utilizaram sementes aleatórias fixas, denominadas tecnicamente *seeds*, para assegurar a reprodutibilidade dos resultados. Por fim, ressalta-se que os valores em negrito apresentados nas tabelas indicam os melhores desempenhos entre os modelos avaliados, ao passo que os valores sublinhados denotam o resultado global superior.

A discussão dos resultados encontra-se estruturada em seis eixos principais:

1. **Análise de desempenho preditivo:** Avalia a eficácia dos modelos nos três níveis de granularidade definidos: Atividade, Operação e Etapa; com o propósito de demonstrar como a complexidade da tarefa influencia de maneira distinta o comportamento de arquiteturas massivas e compactas.

2. **Eficiência computacional:** Quantifica o custo operacional das arquiteturas por meio do isolamento de variáveis críticas, abrangendo o tempo de inferência, o consumo energético e as emissões de carbono, fatores essenciais para determinar a viabilidade de implementação em larga escala.

3. **Análise estatística:** Aplica um rigoroso escrutínio estatístico, mediante testes de hipótese, para validar se as diferenças de desempenho observadas são significativas ou se há equivalência técnica entre os modelos propostos e o estado da arte.

4. **Impacto da KD:** Investiga, de forma isolada, a contribuição da técnica de KD ao analisar como a transferência de conhecimento oriunda do modelo professor atua como um fator de regularização, fenômeno observado especialmente nos problemas de classificação de maior granularidade.

5. **Robustez e estabilidade:** Examina a confiabilidade operacional dos modelos por meio da análise dos coeficientes de variação, com o objetivo de diferenciar as arquiteturas estáveis daquelas que apresentam alto risco de oscilação em ambiente de produção.

6. **Análise ambiental:** Por fim, correlaciona as métricas de precisão e o custo ecológico por meio das Fronteiras de Pareto, identificando as soluções ótimas que maximizam o desempenho por Watt consumido e gCO_2eq/kWh emitido.

4.1 Avaliação comparativa de eficácia

Nesta seção, avalia-se a eficácia preditiva das arquiteturas neurais propostas para a classificação de BDPs. O objetivo central desta etapa consiste em estabelecer o desempenho de referência dos modelos, isolando as métricas de qualidade antes de confrontá-las com os custos computacionais. A análise fundamenta-se nos dados dispostos nas Tabelas 2, 3 e 4, bem como nas Figuras 21, 23 e 25, as quais ilustram o comparativo de F1-Score por modelo.

Tabela 2 – Comparativo das métricas de desempenho preditivo dos modelos avaliados para a classificação de Atividade.

Modelo	Precisão	Revocação	F1-Score
PetroBERT$_{\text{BERTimbau}}$	$0{,}8429 \pm 0{,}0082$	$\mathbf{0{,}8154 \pm 0{,}0026}$	$0{,}8268 \pm 0{,}0034$
PetroBERT$_{\text{mBERT}}$	$\underline{\mathbf{0{,}8489 \pm 0{,}0092}}$	$0{,}8138 \pm 0{,}0068$	$\underline{\mathbf{0{,}8290 \pm 0{,}0057}}$
CNN	$0{,}6358 \pm 0{,}0612$	$0{,}6313 \pm 0{,}0527$	$0{,}6121 \pm 0{,}0543$
CNN$_{\text{KD}}$	$0{,}6407 \pm 0{,}0556$	$0{,}5629 \pm 0{,}0443$	$0{,}5730 \pm 0{,}0535$
BiLSTM	$0{,}7370 \pm 0{,}0209$	$0{,}6760 \pm 0{,}0497$	$0{,}6723 \pm 0{,}0380$
BiLSTM$_{\text{KD}}$	$0{,}8037 \pm 0{,}0419$	$0{,}7730 \pm 0{,}0412$	$0{,}7865 \pm 0{,}0412$
MoE	$0{,}7341 \pm 0{,}0181$	$0{,}7705 \pm 0{,}0147$	$0{,}7430 \pm 0{,}0079$
MoE$_{\text{KD}}$	$0{,}8201 \pm 0{,}0073$	$0{,}7885 \pm 0{,}0171$	$0{,}8030 \pm 0{,}0045$
DistilBERT	$\mathbf{0{,}8429 \pm 0{,}0077}$	$0{,}8128 \pm 0{,}0033$	$\mathbf{0{,}8258 \pm 0{,}0024}$
DistilBERT$_{\text{KD}}$	$0{,}8132 \pm 0{,}0171$	$0{,}8292 \pm 0{,}0048$	$0{,}8180 \pm 0{,}0124$
DistilmBERT	$0{,}8393 \pm 0{,}0053$	$0{,}8008 \pm 0{,}0028$	$0{,}8189 \pm 0{,}0010$
DistilmBERT$_{\text{KD}}$	$0{,}8060 \pm 0{,}0035$	$\underline{\mathbf{0{,}8331 \pm 0{,}0025}}$	$0{,}8159 \pm 0{,}0025$

Fonte: Elaborada pelo autor.

Tabela 3 – Comparativo das métricas de desempenho preditivo dos modelos avaliados para a classificação de Operação.

Modelo	Precisão	Revocação	F1-Score
PetroBERT$_{\text{BERTimbau}}$	$0{,}6755 \pm 0{,}0051$	$\mathbf{0{,}5421 \pm 0{,}0083}$	$\underline{\mathbf{0{,}5773 \pm 0{,}0071}}$
PetroBERT$_{\text{mBERT}}$	$\underline{\mathbf{0{,}6834 \pm 0{,}0079}}$	$0{,}5384 \pm 0{,}0062$	$0{,}5733 \pm 0{,}0060$
CNN	$0{,}2518 \pm 0{,}0311$	$0{,}3118 \pm 0{,}0196$	$0{,}2518 \pm 0{,}0262$
CNN$_{\text{KD}}$	$0{,}3090 \pm 0{,}0481$	$0{,}3346 \pm 0{,}0239$	$0{,}3044 \pm 0{,}0352$
BiLSTM	$0{,}2708 \pm 0{,}0165$	$0{,}2886 \pm 0{,}0163$	$0{,}2480 \pm 0{,}0127$
BiLSTM$_{\text{KD}}$	$0{,}3844 \pm 0{,}0295$	$0{,}3903 \pm 0{,}0172$	$0{,}3681 \pm 0{,}0199$
MoE	$0{,}3847 \pm 0{,}0148$	$0{,}3952 \pm 0{,}0180$	$0{,}3322 \pm 0{,}0073$
MoE$_{\text{KD}}$	$0{,}5104 \pm 0{,}0558$	$0{,}5184 \pm 0{,}0566$	$0{,}4930 \pm 0{,}0629$
DistilBERT	$\mathbf{0{,}6497 \pm 0{,}0231}$	$0{,}5208 \pm 0{,}0208$	$\mathbf{0{,}5563 \pm 0{,}0203}$
DistilBERT$_{\text{KD}}$	$0{,}5386 \pm 0{,}0265$	$\underline{\mathbf{0{,}5539 \pm 0{,}0306}}$	$0{,}5222 \pm 0{,}0267$
DistilmBERT	$0{,}6382 \pm 0{,}0197$	$0{,}5013 \pm 0{,}0205$	$0{,}5382 \pm 0{,}0212$
DistilmBERT$_{\text{KD}}$	$0{,}5530 \pm 0{,}0001$	$0{,}5503 \pm 0{,}0037$	$0{,}5316 \pm 0{,}0024$

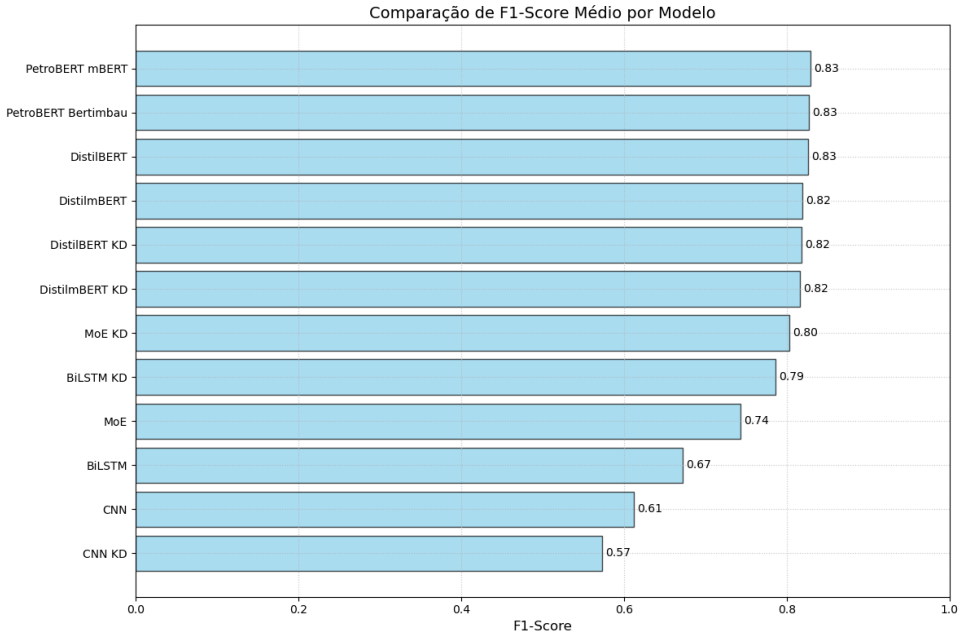
Fonte: Elaborada pelo autor.

Tabela 4 – Comparativo das métricas de desempenho preditivo dos modelos avaliados para a classificação de Etapa.

Modelo	Precisão	Revocação	F1-Score
PetroBERT$_{\text{BERTimbau}}$	$\underline{\mathbf{0{,}5494 \pm 0{,}0081}}$	$\mathbf{0{,}4727 \pm 0{,}0052}$	$\mathbf{0{,}4855 \pm 0{,}0051}$
PetroBERT$_{\text{mBERT}}$	$0{,}5460 \pm 0{,}0071$	$0{,}4644 \pm 0{,}0064$	$0{,}4772 \pm 0{,}0052$
CNN	$0{,}1773 \pm 0{,}0156$	$0{,}2162 \pm 0{,}0076$	$0{,}1799 \pm 0{,}0121$
CNN$_{\text{KD}}$	$0{,}2214 \pm 0{,}0249$	$0{,}2551 \pm 0{,}0166$	$0{,}2260 \pm 0{,}0233$
BiLSTM	$0{,}2844 \pm 0{,}0134$	$0{,}2695 \pm 0{,}0175$	$0{,}2492 \pm 0{,}0150$
BiLSTM$_{\text{KD}}$	$0{,}3733 \pm 0{,}0120$	$0{,}3771 \pm 0{,}0140$	$0{,}3616 \pm 0{,}0138$
MoE	$0{,}4234 \pm 0{,}0012$	$0{,}3822 \pm 0{,}0054$	$0{,}3743 \pm 0{,}0044$
MoE$_{\text{KD}}$	$0{,}4808 \pm 0{,}0083$	$0{,}4383 \pm 0{,}0003$	$0{,}4412 \pm 0{,}0049$
DistilBERT	$0{,}5374 \pm 0{,}0040$	$0{,}4582 \pm 0{,}0056$	$0{,}4722 \pm 0{,}0047$
DistilBERT$_{\text{KD}}$	$0{,}5297 \pm 0{,}0107$	$0{,}4982 \pm 0{,}0157$	$0{,}4954 \pm 0{,}0138$
DistilmBERT	$0{,}5205 \pm 0{,}0027$	$0{,}4683 \pm 0{,}0066$	$0{,}4770 \pm 0{,}0057$
DistilmBERT$_{\text{KD}}$	$\mathbf{0{,}5401 \pm 0{,}0157}$	$\underline{\mathbf{0{,}5067 \pm 0{,}0076}}$	$\underline{\mathbf{0{,}5023 \pm 0{,}0024}}$

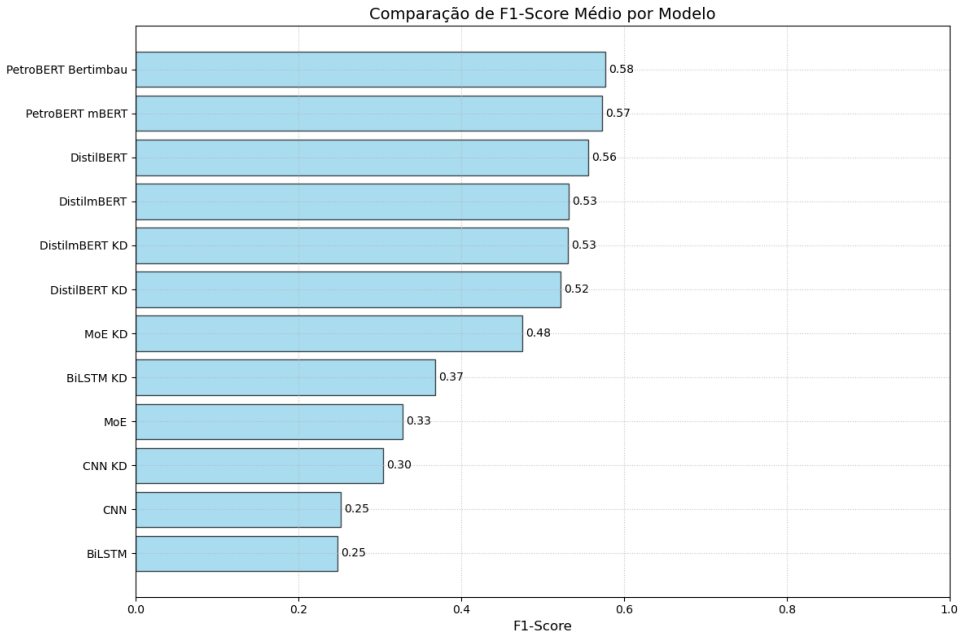
Fonte: Elaborada pelo autor.

Figura 14 – Comparação visual do F1-Score médio por modelo para a classificação de Atividade.



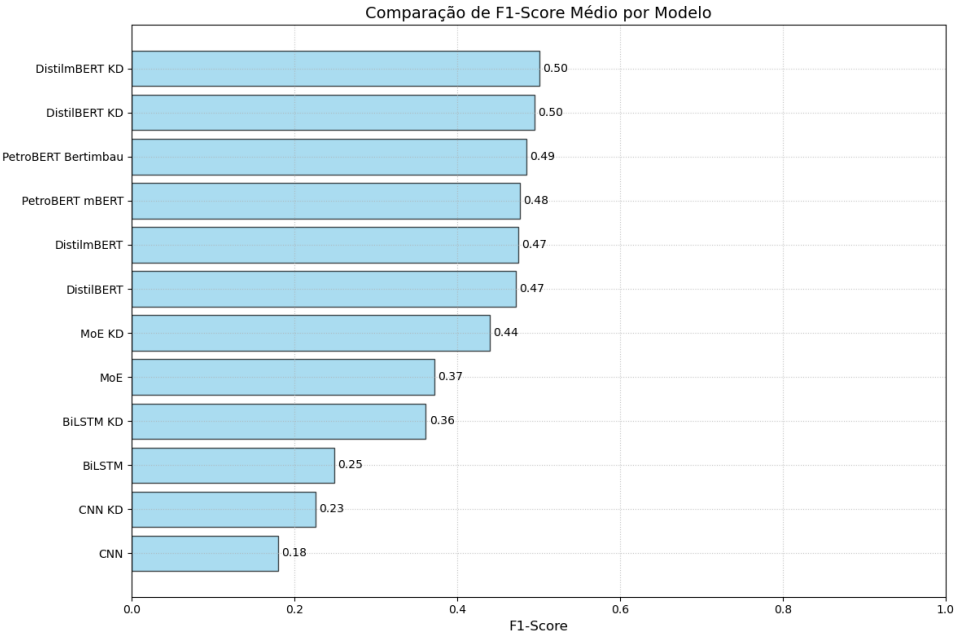
Fonte: Elaborada pelo autor.

Figura 15 – Comparação visual do F1-Score médio por modelo para a classificação de Operação.



Fonte: Elaborada pelo autor.

Figura 16 – Comparação visual do F1-Score médio por modelo para a classificação de Etapa.



Fonte: Elaborada pelo autor.

Os experimentos evidenciam um comportamento heterogêneo dos modelos frente ao aumento da complexidade da tarefa, com o nível de Atividade apresentando os melhores resultados globais. Nesse cenário, as arquiteturas baseadas em BERT predominam no topo da avaliação. O modelo PetroBERT$_{\text{mBERT}}$ obteve o melhor F1-Score geral ao registrar $0{,}8290 \pm 0{,}0057$, resultado que configura um empate estatístico com o PetroBERT$_{\text{BERTimbau}}$, o qual alcançou a marca de $0{,}8268$. Nota-se que as versões compactas, tanto as não destiladas quanto as submetidas à KD, especificamente o DistilBERT e o DistilmBERT, com pontuações de $0{,}8258$ e $0{,}8189$, respectivamente, mantêm uma performance notavelmente próxima à de seus professores, o que representa uma perda marginal de desempenho inferior a 1%. Em contrapartida, arquiteturas tradicionais como CNN e BiLSTM apresentam resultados significativamente inferiores, fato exemplificado pela CNN padrão que atingiu apenas $0{,}6121$ de F1-Score. Contudo, a aplicação de KD nesses modelos menores gerou ganhos expressivos e permitiu que o BiLSTM$_{\text{KD}}$ elevasse seu desempenho para $0{,}7865$, aproximando-se assim do patamar estabelecido pelos modelos BERT.

Com o aprofundamento da granularidade para o nível de Operação, observa-se uma degradação natural das métricas em todos os modelos avaliados. O PetroBERT$_{\text{BERTimbau}}$ lidera o comparativo ao registrar um F1-Score de $0{,}5773$. Nesse patamar, a distinção hierárquica entre modelos robustos e leves torna-se mais tênue, embora a discrepância numérica entre eles se acentue também. O DistilBERT demonstra manter sua competitividade ao atingir a marca de $0{,}5563$. Um aspecto crucial reside no desempenho das arquiteturas não fundamentadas em BERT; mesmo com a aplicação de KD, o MoE$_{\text{KD}}$ e o BiLSTM$_{\text{KD}}$, cujos índices foram de $0{,}4930$ e $0{,}3681$ respectivamente, mostram-se incapazes de rivalizar com as abordagens

baseadas em mecanismos de atenção. Tal cenário sugere que a complexidade semântica inerente à classificação de Operações demanda a capacidade de modelagem de contexto característica da família BERT.

Por fim, ao se examinar o nível de maior granularidade e complexidade, identifica-se uma inversão nos padrões de desempenho. O modelo comprimido DistilmBERT$_{\text{KD}}$ alcançou o maior F1-Score, com $0{,}5023 \pm 0{,}0024$, e a maior revocação, de $0{,}5067$, superando inclusive as arquiteturas massivas PetroBERT$_{\text{BERTimbau}}$ e PetroBERT$_{\text{mBERT}}$, as quais registraram $0{,}4855$ e $0{,}4772$, respectivamente. Esse resultado sugere que, para classes de elevada especificidade, a regularização induzida pelo processo de destilação pode ter mitigado o sobreajuste e permitido que o modelo compacto apresentasse uma capacidade de generalização superior àquela demonstrada pelo modelo professor.

As figuras elucidam os agrupamentos de desempenho e o impacto das técnicas de compressão. Ao se examinar a Figura 14, identifica-se um platô de alto desempenho na região superior, onde os modelos PetroBERT, DistilBERT, DistilmBERT, suas respectivas variantes destiladas e o MoE$_{\text{KD}}$ constituem um bloco coeso com resultados superiores a $0{,}80$. Em contrapartida, o gráfico evidencia um declínio acentuado de performance para as arquiteturas BiLSTM e CNN, as quais se posicionam na porção inferior da distribuição. Além disso, observa-se que a técnica de KD, identificada pelas barras com o sufixo correspondente, demonstra eficácia ao impulsionar os resultados de modelos intermediários, a exemplo do MoE e do BiLSTM, reduzindo a disparidade em relação aos líderes do ranking.

Por sua vez, a Figura 15 ilustra um aumento na dispersão dos resultados. Enquanto as arquiteturas com melhor desempenho sustentam-se acima de $0{,}50$, aquelas situadas no limite inferior, especificamente a CNN e a BiLSTM, apresentam uma redução drástica, situando-se no intervalo entre $0{,}20$ e $0{,}30$. Essa análise evidencia que, na ausência de mecanismos de atenção, a classificação neste nível de complexidade torna-se impraticável, independentemente da aplicação de destilação. Tal limitação é corroborada pela diferença de $0{,}10$ observada no MoE$_{\text{KD}}$ em relação ao *baseline*.

Finalmente, a Figura 16 corrobora os achados numéricos da Tabela 4. Observa-se uma acentuada proximidade entre as barras superiores, cenário em que o DistilmBERT$_{\text{KD}}$ e o DistilBERT$_{\text{KD}}$, ambos situados no patamar de $0{,}50$, se equiparam visualmente ou até superam os modelos PetroBERT. Tal configuração sugere que, para a tarefa de maior granularidade, o custo computacional excedente das arquiteturas robustas não se traduz em ganhos proporcionais de desempenho, o que torna questionável a justificativa para sua utilização nesse contexto específico.

Ademais, a redução monotonicamente decrescente do F1-Score, variando de aproximadamente $0{,}83$ no nível de Atividade para cerca de $0{,}50$ no nível de Etapa, reflete a ambiguidade intrínseca aos registros operacionais. À medida que as classes se tornam mais específicas e numerosas, a distinção léxica diminui, impondo ao modelo a necessidade de capturar nuances

sutis nem sempre presentes nos textos curtos dos BDPs. Tal limitação elucida a dificuldade generalizada enfrentada por todos os modelos nas etapas de maior granularidade.

4.2 Caracterização do custo de inferência e consumo de recursos

Nesta seção, o escopo da análise transita da eficácia preditiva para a eficiência computacional, pilar central das diretrizes de Green AI e requisito fundamental para a viabilidade industrial. O objetivo consiste em quantificar o custo operacional das arquiteturas avaliadas ao analisar as variáveis de consumo de recursos independentemente do desempenho obtido em F1-Score. A discussão fundamenta-se nos dados apresentados nas Tabelas 5, 6 e 7. Nesse contexto, examinam-se métricas críticas que abrangem a emissão de gCO_2eq/kWh, o consumo energético em quilowatts-hora (kWh), o tempo de inferência em segundos (s), as operações de ponto flutuante em tera-operações (T) e a quantidade absoluta de parâmetros dos modelos.

Tabela 5 – Métricas de eficiência computacional e impacto ambiental dos modelos para o nível de Atividade.

Modelo	**gCO_2eq/kWh**	**Energia (kWh)**	**Tempo de Inferência (s)**	**FLOPs (T)**	**Parâmetros (M)**
$PetroBERT_{BERTimbau}$	$0{,}8375 \pm 0{,}0269$	$0{,}0085 \pm 0{,}0003$	$201{,}7320 \pm 2{,}0092$	23,71	108,93
$PetroBERT_{mBERT}$	$0{,}8347 \pm 0{,}0166$	$0{,}0085 \pm 0{,}0002$	$202{,}1720 \pm 2{,}5572$	23,71	177,86
CNN	$0{,}0165 \pm 0{,}0022$	$0{,}0002 \pm 0{,}0000$	$0{,}7963 \pm 0{,}0005$	0,16	8,25
CNN_{KD}	$0{,}0159 \pm 0{,}0011$	$0{,}0002 \pm 0{,}0000$	$0{,}7971 \pm 0{,}0006$	0,16	8,25
BiLSTM	$0{,}0111 \pm 0{,}0023$	$0{,}0001 \pm 0{,}0000$	$1{,}0080 \pm 0{,}0040$	0,52	9,60
$BiLSTM_{KD}$	$0{,}0103 \pm 0{,}0008$	$0{,}0001 \pm 0{,}0000$	$1{,}0040 \pm 0{,}0049$	0,52	9,60
MoE	$0{,}0954 \pm 0{,}0015$	$0{,}0010 \pm 0{,}0000$	$10{,}1180 \pm 0{,}0040$	5,16	27,41
MoE_{KD}	$0{,}0960 \pm 0{,}0020$	$0{,}0010 \pm 0{,}0000$	$10{,}1000 \pm 0{,}0000$	5,16	27,41
DistilBERT	$0{,}4449 \pm 0{,}0021$	$0{,}0045 \pm 0{,}0000$	$23{,}3320 \pm 0{,}0040$	11,86	65,79
$DistilBERT_{KD}$	$0{,}4436 \pm 0{,}0013$	$0{,}0045 \pm 0{,}0000$	$23{,}3600 \pm 0{,}0110$	11,86	65,79
DistilmBERT	$0{,}4440 \pm 0{,}0017$	$0{,}0045 \pm 0{,}0000$	$23{,}3710 \pm 0{,}0153$	11,86	135,33
$DistilmBERT_{KD}$	$0{,}4442 \pm 0{,}0023$	$0{,}0045 \pm 0{,}0000$	$23{,}3950 \pm 0{,}0071$	11,86	135,33

Fonte: Elaborada pelo autor.

Tabela 6 – Métricas de eficiência computacional e impacto ambiental dos modelos para o nível de Operação.

Modelo	gCO_2eq/kWh	Energia (kWh)	Tempo de Inferência (s)	FLOPs (T)	Parâmetros (M)
$PetroBERT_{BERTimbau}$	$0{,}8321 \pm 0{,}0227$	$0{,}0085 \pm 0{,}0002$	$201{,}0080 \pm 1{,}4207$	23,71	108,98
$PetroBERT_{mBERT}$	$0{,}8396 \pm 0{,}0164$	$0{,}0085 \pm 0{,}0002$	$203{,}6120 \pm 2{,}6903$	23,71	177,91
CNN	$0{,}0157 \pm 0{,}0008$	$0{,}0002 \pm 0{,}0000$	$0{,}7965 \pm 0{,}0008$	0,16	8,29
CNN_{KD}	$0{,}0160 \pm 0{,}0010$	$0{,}0002 \pm 0{,}0000$	$0{,}7978 \pm 0{,}0003$	0,16	8,29
BiLSTM	$0{,}0104 \pm 0{,}0010$	$0{,}0001 \pm 0{,}0000$	$1{,}0100 \pm 0{,}0000$	0,52	9,66
$BiLSTM_{KD}$	$0{,}0105 \pm 0{,}0007$	$0{,}0001 \pm 0{,}0000$	$1{,}0100 \pm 0{,}0000$	0,52	9,66
MoE	$0{,}0953 \pm 0{,}0008$	$0{,}0010 \pm 0{,}0000$	$10{,}0950 \pm 0{,}0050$	5,16	27,96
MoE_{KD}	$0{,}0956 \pm 0{,}0011$	$0{,}0010 \pm 0{,}0000$	$10{,}1300 \pm 0{,}0000$	5,16	27,96
DistilBERT	$0{,}4443 \pm 0{,}0016$	$0{,}0045 \pm 0{,}0000$	$23{,}3260 \pm 0{,}0049$	11,86	65,84
$DistilBERT_{KD}$	$0{,}4439 \pm 0{,}0016$	$0{,}0045 \pm 0{,}0000$	$23{,}4820 \pm 0{,}0172$	11,86	65,84
DistilmBERT	$0{,}4457 \pm 0{,}0028$	$0{,}0045 \pm 0{,}0000$	$23{,}5600 \pm 0{,}0283$	11,86	135,39
$DistilmBERT_{KD}$	$0{,}4447 \pm 0{,}0026$	$0{,}0045 \pm 0{,}0000$	$23{,}4550 \pm 0{,}0354$	11,86	135,39

Fonte: Elaborada pelo autor.

Tabela 7 – Métricas de eficiência computacional e impacto ambiental dos modelos para o nível de Etapa.

Modelo	gCO_2eq/kWh	Energia (kWh)	Tempo de Inferência (s)	FLOPs (T)	Parâmetros (M)
$PetroBERT_{BERTimbau}$	$0{,}8408 \pm 0{,}0247$	$0{,}0085 \pm 0{,}0003$	$202{,}7220 \pm 1{,}4348$	23,71	109,12
$PetroBERT_{mBERT}$	$0{,}8373 \pm 0{,}0159$	$0{,}0085 \pm 0{,}0002$	$203{,}5080 \pm 2{,}1273$	23,71	178,05
CNN	$0{,}0157 \pm 0{,}0007$	$0{,}0002 \pm 0{,}0000$	$0{,}7969 \pm 0{,}0004$	0,16	8,39
CNN_{KD}	$0{,}0158 \pm 0{,}0008$	$0{,}0002 \pm 0{,}0000$	$0{,}7970 \pm 0{,}0005$	0,16	8,39
BiLSTM	$0{,}0105 \pm 0{,}0010$	$0{,}0001 \pm 0{,}0000$	$1{,}0100 \pm 0{,}0000$	0,52	9,79
$BiLSTM_{KD}$	$0{,}0103 \pm 0{,}0007$	$0{,}0001 \pm 0{,}0000$	$1{,}0100 \pm 0{,}0000$	0,52	9,79
MoE	$0{,}0961 \pm 0{,}0023$	$0{,}0010 \pm 0{,}0000$	$10{,}1470 \pm 0{,}0051$	5,17	29,31
MoE_{KD}	$0{,}0954 \pm 0{,}0012$	$0{,}0010 \pm 0{,}0000$	$10{,}1600 \pm 0{,}0000$	5,17	29,31
DistilBERT	$0{,}4445 \pm 0{,}0011$	$0{,}0045 \pm 0{,}0000$	$23{,}4020 \pm 0{,}0075$	11,86	65,98
$DistilBERT_{KD}$	$0{,}4439 \pm 0{,}0012$	$0{,}0045 \pm 0{,}0000$	$23{,}5320 \pm 0{,}0098$	11,86	65,98
DistilmBERT	$0{,}4449 \pm 0{,}0030$	$0{,}0045 \pm 0{,}0000$	$23{,}5040 \pm 0{,}0612$	11,86	135,52
$DistilmBERT_{KD}$	$0{,}4446 \pm 0{,}0020$	$0{,}0045 \pm 0{,}0000$	$23{,}5170 \pm 0{,}0051$	11,86	135,52

Fonte: Elaborada pelo autor.

Os dados evidenciam uma disparidade de magnitude entre as famílias de modelos, tendência que permanece constante independentemente do nível de granularidade da classificação. As arquiteturas fundamentadas no BERT estabelecem o teto de consumo de recursos. Nesses casos, a latência média oscila entre 201 e 203 s, associada a um consumo energético de $0{,}0085$ kWh e a emissões de gCO_2eq/kWh superiores a $0{,}83$ por inferência. O volume de FLOPs atinge a expressiva marca de $23{,}71$ T, fator determinante para o elevado tempo de processamento verificado.

Em contrapartida, o DistilBERT e suas variantes apresentaram uma latência média de

aproximadamente $23{,}3$ s, o que corresponde a uma aceleração de quase nove vezes em relação ao modelo professor. O consumo de energia e as emissões de gCO_2eq/kWh também sofreram redução da ordem de 50%, fixando-se em cerca de $0{,}0045$ kWh e $0{,}44$, respectivamente. Observa-se que, embora a quantidade de parâmetros varie entre 65 M para o DistilBERT e 135 M para o DistilmBERT, o tempo de inferência manteve-se similar. Tal comportamento sugere que a profundidade da rede, definida pelo número de camadas, exerce influência mais determinante sobre a latência do que a largura da camada de *embeddings* no cenário analisado.

Por sua vez, os modelos não baseados em BERT representam o extremo da eficiência computacional. As arquiteturas CNN e BiLSTM operam com latências da ordem de $0{,}79$ s e $1{,}01$ s, respectivamente, o que corresponde a uma velocidade de processamento cerca de 200 vezes superior à do PetroBERT. Nesse cenário, o consumo energético mostra-se praticamente desprezível, situando-se entre $0{,}0001$ e $0{,}0002$ kWh, ao passo que as emissões de gCO_2eq/kWh equivalente são mínimas, com valores próximos a $0{,}01$. Já o modelo MoE ocupa uma posição intermediária, com latência de aproximadamente 10 s e emissões em torno de $0{,}09$, oferecendo, assim, um equilíbrio entre as arquiteturas LSTM e CNN e aquelas da família BERT.

A análise comparativa das três tabelas revela que a variação nos indicadores de eficiência entre os níveis de Atividade, Operação e Etapa mostra-se marginal, restringindo-se às casas decimais. A título de exemplo, o número de parâmetros do $\text{PetroBERT}_{\text{BERTimbau}}$ eleva-se de $108{,}93$ M para apenas $109{,}12$ M. Esse comportamento confirma que o custo computacional é dominado pelo *backbone* da rede neural em detrimento da camada de classificação final, tecnicamente denominada *head*, cujo aumento dimensional necessário para acomodar um maior número de classes exerce impacto negligenciável sobre o custo total.

Os resultados validam a hipótese de que a compressão de modelos via destilação não constitui apenas uma otimização teórica, mas sim uma necessidade concreta para a viabilidade da implantação de sistemas. A técnica de KD não adicionou *overhead* computacional aos modelos alunos, fato observável ao se compararem a CNN e a CNN_{KD}, cujos tempos de execução se mostraram estatisticamente idênticos. Tal evidência comprova que o custo associado à KD restringe-se exclusivamente à fase de treinamento, denominada *offline*, o que resulta em uma etapa de inferência, ou *online*, isenta de custos adicionais.

4.3 Validação estatística de equivalência e superioridade

Nesta etapa de avaliação, os resultados de desempenho são submetidos a um rigoroso escrutínio estatístico com o intuito de determinar a significância das diferenças observadas entre o modelo *baseline* PetroBERT e as arquiteturas propostas. O objetivo primordial desta análise consiste em validar se a redução da complexidade computacional acarreta uma perda de precisão estatisticamente relevante ou se os modelos comprimidos podem ser considerados equivalentes ao estado da arte. A fundamentação deste estudo baseia-se nos dados apresentados

nas Tabelas 8, 9 e 10, utilizando-se o teste-t pareado com a correção de Holm-Bonferroni e a métrica d de Cohen para mensurar a magnitude do efeito da diferença.

Com o objetivo de assegurar a robustez das inferências sobre a equivalência ou a superioridade dos modelos propostos em relação ao *baseline*, procedeu-se a uma análise estatística rigorosa dos resultados obtidos nas validações cruzadas. Preliminarmente, a aderência dos dados de desempenho referentes ao F1-Score à distribuição gaussiana foi verificada por meio do teste de normalidade de Shapiro-Wilk. Os resultados indicaram que a distribuição majoritária das amostras não viola a premissa de normalidade ao apresentar um valor de $p > 0{,}05$, fato que fundamenta a utilização de testes paramétricos para a comparação das médias.

Nesse contexto, adotou-se o teste-t pareado como o principal instrumento para a verificação de hipóteses. A seleção dessa variante justifica-se pelo desenho experimental estabelecido, no qual os distintos modelos foram avaliados sobre as mesmas instâncias de teste, fato que configura uma dependência direta entre as observações. Consequentemente, o teste avaliou a hipótese nula H_0, a qual postula que a diferença média de desempenho entre o modelo *baseline* PetroBERT$_{\text{BERTimbau}}$ e as arquiteturas compactas é igual a zero, enquanto a hipótese alternativa H_1 pressupõe a existência de uma diferença estatisticamente significativa entre as médias comparadas.

Tendo em vista que a validação implica o confronto simultâneo de múltiplos modelos contra um único *baseline*, eleva-se o risco de ocorrência de erros do Tipo I, tecnicamente denominados falsos positivos. Para mitigar a obtenção de conclusões espúrias decorrentes desse fenômeno, aplicou-se a correção de Holm-Bonferroni aos valores de p. Tal método ajusta o nível de significância de forma sequencial e é reconhecido por oferecer um controle eficaz da Taxa de Erro Familiar, ou Family-Wise Error Rate (FWER) em inglês, destacando-se por ser menos conservador e por apresentar maior poder estatístico em comparação à correção de Bonferroni tradicional.

Por fim, reconhecendo-se que a significância estatística expressa pelo valor de p não implica necessariamente relevância prática, procedeu-se ao cálculo da métrica de tamanho de efeito d de Cohen. Esse coeficiente quantifica a magnitude da diferença padronizada entre as médias de desempenho dos modelos, permitindo uma avaliação mais robusta da utilidade das arquiteturas propostas. Para a interpretação dos resultados, adotaram-se os limiares convencionais. Dessa forma, torna-se possível distinguir as diferenças que, embora estatisticamente significativas devido à dimensão da amostra, podem se revelar negligenciáveis sob a ótica da aplicação real na engenharia de petróleo.

Tabela 8 – Resultados do teste-t pareado e tamanho de efeito comparando os modelos propostos ao *baseline* no nível de Atividade.

Modelo	p-valor	d de Cohen	Hipótese	Efeito
CNN	0,0091	4,9913	Rejeitada	Grande
CNN_{KD}	0,0067	5,9879	Rejeitada	Grande
BiLSTM	0,0076	5,1188	Rejeitada	Grande
$BiLSTM_{KD}$	0,4237	1,2318	Aceita	Grande
MoE	0,0004	12,3245	Rejeitada	Grande
MoE_{KD}	0,0091	5,3738	Rejeitada	Grande
DistilBERT	0,7435	0,2968	Aceita	Pequeno
$DistilBERT_{KD}$	0,5972	0,8676	Aceita	Grande
DistilmBERT	0,0701	2,8291	Aceita	Grande
$DistilmBERT_{KD}$	0,0409	3,2629	Rejeitada	Grande

Fonte: Elaborada pelo autor.

Tabela 9 – Resultados do teste-t pareado e tamanho de efeito comparando os modelos propostos ao *baseline* no nível de Operação.

Modelo	p-valor	d de Cohen	Hipótese	Efeito
CNN	0,0002	15,1443	Rejeitada	Grande
CNN_{KD}	0,0006	9,6113	Rejeitada	Grande
BiLSTM	0,0000	28,6557	Rejeitada	Grande
$BiLSTM_{KD}$	0,0006	12,5148	Rejeitada	Grande
MoE	0,0000	25,2825	Rejeitada	Grande
MoE_{KD}	0,0210	2,8159	Rejeitada	Grande
DistilBERT	0,0600	1,2329	Aceita	Grande
$DistilBERT_{KD}$	0,0197	2,5196	Rejeitada	Grande
DistilmBERT	0,0197	3,3795	Rejeitada	Grande
$DistilmBERT_{KD}$	0,0010	8,0375	Rejeitada	Grande

Fonte: Elaborada pelo autor.

Tabela 10 – Resultados do teste-t pareado e tamanho de efeito comparando os modelos propostos ao *baseline* no nível de Etapa.

Modelo	p-valor	d de Cohen	Hipótese	Efeito
CNN	0,0000	29,4073	Rejeitada	Grande
CNN_{KD}	0,0001	13,7417	Rejeitada	Grande
BiLSTM	0,0001	18,9112	Rejeitada	Grande
$BiLSTM_{KD}$	0,0005	10,6673	Rejeitada	Grande
MoE	0,0001	20,8928	Rejeitada	Grande
MoE_{KD}	0,0009	9,2972	Rejeitada	Grande
DistilBERT	0,1472	2,4108	Aceita	Grande
$DistilBERT_{KD}$	0,2192	−0,8460	Aceita	Grande
DistilmBERT	0,1472	1,4469	Aceita	Grande
$DistilmBERT_{KD}$	0,0192	−3,2718	Rejeitada	Grande

Fonte: Elaborada pelo autor.

Os testes de hipótese revelam cenários distintos de equivalência que variam conforme o nível hierárquico e a família do modelo analisado. No nível de Atividade, observa-se que o modelo DistilBERT obteve um valor de p igual a $0{,}7435$, o que resultou na aceitação da hipótese nula de equivalência em relação ao *baseline*, comportamento igualmente verificado para o DistilBERT$_{\text{KD}}$ e para o DistilmBERT. Tal desfecho é corroborado por um d de Cohen de $0{,}2968$ para o primeiro modelo, valor classificado como um efeito pequeno que indica uma sobreposição quase total das distribuições de desempenho, ao passo que os demais apresentaram efeito grande. De maneira surpreendente, a BiLSTM$_{\text{KD}}$ também teve a hipótese de equivalência aceita com um valor de p de $0{,}4237$, embora apresente um tamanho de efeito grande, com d igual a $1{,}23$, o que sugere uma alta variância nos resultados. Em contrapartida, os modelos clássicos sem destilação, como CNN e MoE, tiveram a hipótese de equivalência rejeitada ao apresentarem valores de p críticos inferiores a $0{,}01$ e efeitos grandes, com d superior a $4{,}9$.

No cenário de Operação, caracterizado por sua maior complexidade, verifica-se que a maioria dos modelos não sustenta a equivalência estatística em relação ao *baseline*. As arquiteturas recorrentes e convolucionais apresentam valores de d de Cohen elevados, variando de $9{,}6$ a $28{,}6$, índices que denotam uma degradação de desempenho expressiva e inquestionável. A única exceção reside no DistilBERT, que permanece estatisticamente equivalente ao modelo professor com um valor de p igual a $0{,}0600$, resultado situado no limiar da significância ao se considerar um nível de significância de $0{,}05$. Tal métrica é acompanhada por um tamanho de efeito grande, com d igual a $1{,}23$, o que indica que, embora a hipótese nula não tenha sido rejeitada, existe uma distância prática perceptível entre o desempenho dos modelos comparados.

Por fim, ao se analisar o nível de Etapa, identificam-se novamente os fenômenos estatísticos de inversão. Os modelos DistilBERT e DistilBERT$_{\text{KD}}$, ao registrarem valores de p iguais a $0{,}1472$ e $0{,}2192$ respectivamente, tiveram suas hipóteses de equivalência aceitas. Um aspecto crucial dessa análise refere-se aos valores negativos do d de Cohen observados nos modelos destilados, atingindo $-0{,}8460$ para o DistilBERT$_{\text{KD}}$ e $-3{,}2718$ para o DistilmBERT$_{\text{KD}}$. No caso específico do DistilmBERT$_{\text{KD}}$, a hipótese de equivalência foi rejeitada com um valor de p de $0{,}0192$. Entretanto, a magnitude negativa do efeito confirma que a diferença observada é significativa em favor do modelo comprimido, o que demonstra sua superioridade estatística em relação ao *baseline*.

A visualização tabular dos dados estatísticos permite identificar três agrupamentos distintos de comportamento. O primeiro constitui um *cluster* de rejeição robusta, formado pelas arquiteturas CNN, BiLSTM e MoE, considerando-se tanto as versões originais quanto aquelas submetidas à aplicação de KD. Independentemente do nível analisado, esses modelos apresentam valores de p próximos a zero e efeitos de magnitude extrema, com o coeficiente d superior a 5. A consistência desses resultados entre as tabelas indica que a inferioridade de tais modelos não decorre de variações aleatórias, mas reflete uma limitação arquitetural intrínseca para a tarefa em questão.

Identifica-se também um *cluster* de equivalência que estabelece uma zona de Pareto, no qual o DistilBERT puramente pré-treinado figura como o único modelo a manter o status de hipótese aceita em todos os três níveis hierárquicos. Essa característica é evidenciada pelos baixos valores de d de Cohen apresentados na Tabela 8 e pela resiliência do valor de p acima de $0{,}05$ nas demais avaliações.

Por fim, observa-se um *cluster* de inversão de desempenho na tabela referente à Etapa, onde os valores negativos de d de Cohen para os modelos destilados sinalizam uma ruptura no padrão. Esse comportamento demarca o ponto em que a compressão deixa de ser uma concessão de performance para se tornar uma vantagem de generalização.

4.4 Quantificação do ganho por destilação de conhecimento

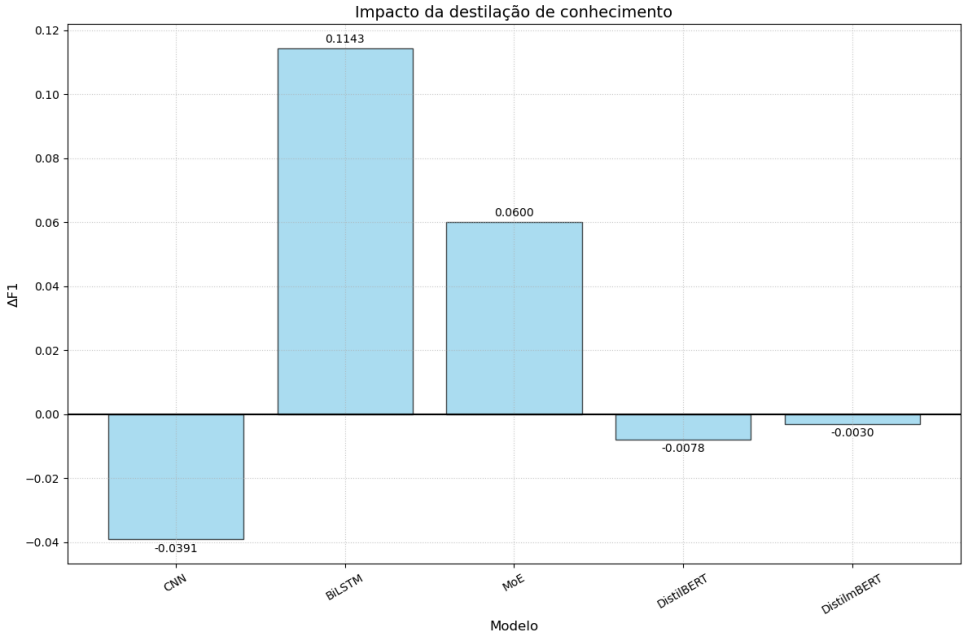
Esta seção dedica-se a isolar e quantificar a contribuição específica da técnica de KD no desempenho preditivo das arquiteturas avaliadas. O objetivo primordial consiste em verificar se, e em que proporção, a transferência de conhecimento do modelo professor, aqui representado pelo PetroBERT$_{\text{BERTimbau}}$, para os modelos alunos resulta em ganhos de generalização que justifiquem o custo adicional de treinamento. A análise fundamenta-se na Tabela 11, que consolida os valores absolutos de F1-Score obtidos pelos modelos ao confrontar os cenários com e sem a aplicação de KD e a variação correspondente indicada como $\Delta F1$. Adicionalmente, a discussão baseia-se nas figuras que ilustram o ganho ou a perda de desempenho por arquitetura.

Tabela 11 – Quantificação do ganho ou perda de desempenho obtido através da técnica de KD nos três níveis hierárquicos.

Arquitetura	**Nível de Classificação**								
	Atividade			**Operação**			**Etapa**		
	F1	**F1$_{KD}$**	**ΔF1**	**F1**	**F1$_{KD}$**	**ΔF1**	**F1**	**F1$_{KD}$**	**ΔF1**
CNN	0,6121	0,5730	−0,0391	0,2518	0,3044	0,0526	0,1799	0,2260	0,0461
BiLSTM	0,6723	0,8037	0,1143	0,2480	0,3681	0,1201	0,2492	0,3616	0,1124
MoE	0,7430	0,8030	0,0600	0,3322	0,4930	0,1470	0,3743	0,4412	0,0672
DistilBERT	0,8258	0,8180	−0,0078	0,5563	0,5222	−0,0341	0,4722	0,4954	0,0231
DistilmBERT	0,8189	0,8159	−0,0030	0,5382	0,5316	−0,0013	0,4770	0,5023	0,0266

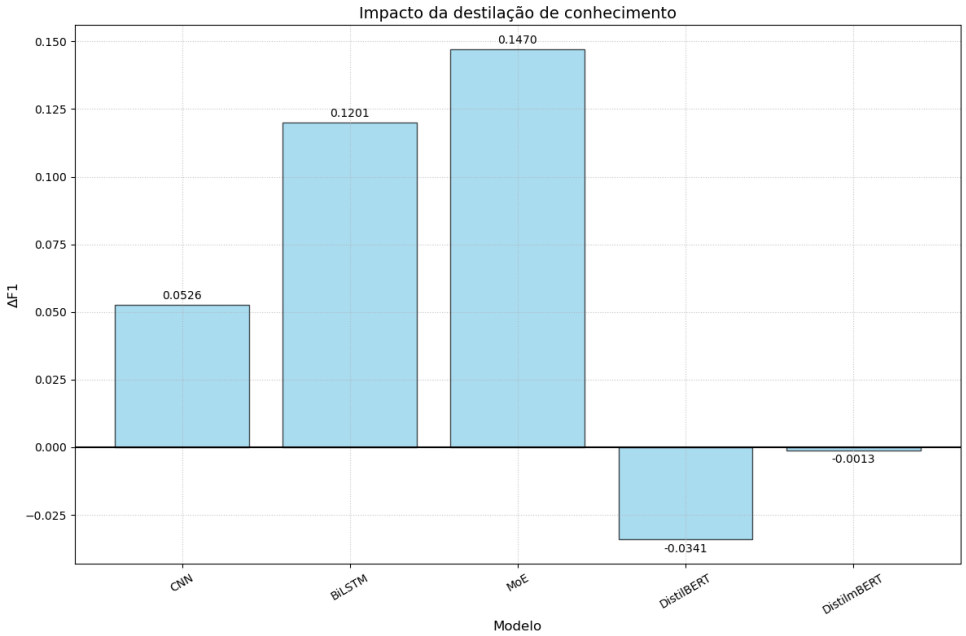
Fonte: Elaborada pelo autor.

Figura 17 – Impacto da KD no desempenho preditivo para o nível de Atividade.



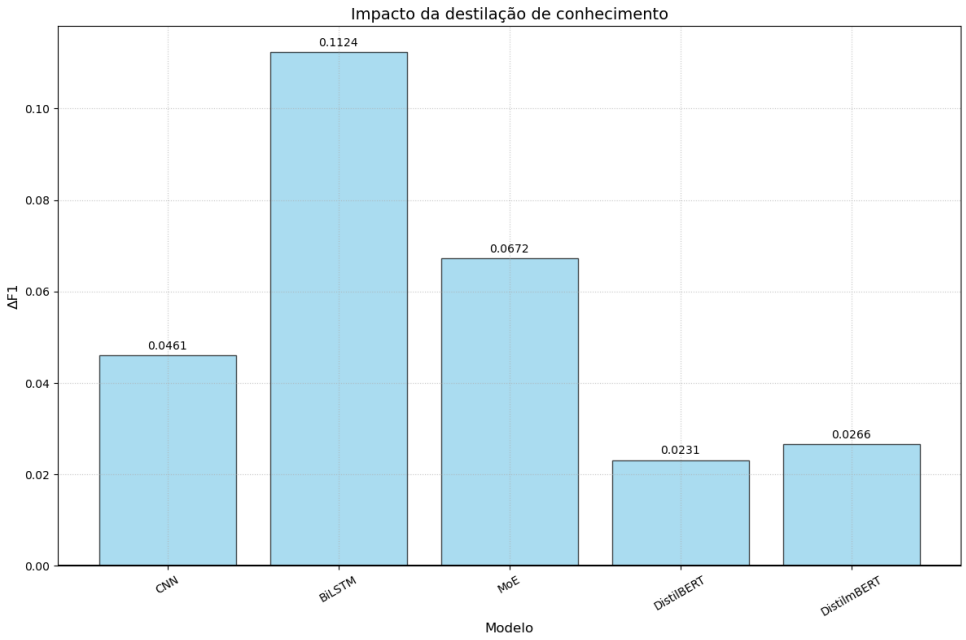
Fonte: Elaborada pelo autor.

Figura 18 – Impacto da KD no desempenho preditivo para o nível de Operação.



Fonte: Elaborada pelo autor.

Figura 19 – Impacto da KD no desempenho preditivo para o nível de Etapa.



Fonte: Elaborada pelo autor.

A Tabela 11 revela um comportamento heterogêneo da KD, o qual se mostra fortemente correlacionado tanto à arquitetura do modelo aluno quanto à complexidade da tarefa. Nesse panorama, o modelo BiLSTM apresentou os ganhos mais consistentes e expressivos em todos os cenários avaliados, registrando incrementos de $\Delta F1$ iguais a $0{,}1143$ no nível de Atividade, $0{,}1201$ em Operação e $0{,}1124$ em Etapa. O modelo MoE seguiu uma tendência similar, com destaque para o nível de Operação, no qual a destilação impulsionou o F1-Score de $0{,}3322$ para $0{,}4930$. Esse salto de $0{,}1470$ representa o maior ganho registrado neste estudo. Por outro lado, a arquitetura CNN demonstrou instabilidade, evidenciada por uma degradação de desempenho no nível de Atividade equivalente a $-0{,}0391$. Contudo, nos níveis subsequentes de maior granularidade, a técnica mostrou-se benéfica, resultando em ganhos de $0{,}0526$ e $0{,}0461$ para Operação e Etapa, respectivamente.

Já no que tange aos modelos baseados em atenção, especificamente o DistilBERT e o DistilmBERT, a destilação apresentou retornos marginais ou negativos nos níveis hierárquicos superiores. Em Atividade e Operação, registraram-se perdas leves, exemplificadas pela redução de $0{,}0341$ no DistilBERT para o nível de Operação. Entretanto, observa-se uma inversão crucial no nível de Etapa, considerado o mais complexo, no qual ambos os modelos obtiveram ganhos positivos com $\Delta F1$ de aproximadamente $0{,}02$. Esse resultado indica que a destilação auxiliou na desambiguação de classes mais finas, favorecendo a generalização em tarefas de maior granularidade.

As representações gráficas permitem visualizar a magnitude do impacto e a inversão de tendências observadas no experimento. No que tange ao nível de Atividade, o gráfico é

dominado pela barra positiva referente à BiLSTM, a qual ultrapassa o valor de $0{,}10$ e estabelece um contraste com o desempenho negativo da CNN. Visualmente, nota-se que, para tarefas de alto nível, modelos que já possuem alta capacidade, como as variantes BERT, não obtêm benefícios significativos com a aplicação de KD ao apresentarem variações próximas a zero ou levemente negativas. Já em relação ao nível de Operação, a disparidade se acentua, uma vez que os resultados da BiLSTM e do MoE indicam ser a destilação fundamental para que esses modelos atinjam um patamar aceitável de performance.

O DistilBERT, por sua vez, apresenta sua retração mais acentuada, o que sugere um conflito entre o aprendizado supervisionado direto e a imitação do professor nesse nível específico. Por fim, a análise das Etapas exibe um padrão visual singular no qual todas as barras são positivas, sem registros abaixo do eixo zero. Tal comportamento ilustra que, à medida que a tarefa se torna mais complexa e ruidosa, a orientação do professor torna-se universalmente benéfica, independentemente da arquitetura do modelo aluno.

Os resultados obtidos corroboram a teoria da lacuna de capacidade abordada por Zhang et al. (2023b) no contexto da KD. Observa-se que arquiteturas dotadas de viés indutivo forte e menor capacidade de modelagem de dependências de longo alcance, a exemplo da BiLSTM e do MoE, extraem o máximo proveito da KD. O conhecimento obscuro contido nos *logits* do professor fornece relações entre classes que esses modelos não conseguiriam assimilar apenas por meio dos rótulos rígidos, os quais são comumente representados por vetores *one-hot*. Especificamente para a BiLSTM, a destilação atuou como uma correção de arquitetura, o que permitiu ao modelo emular o mecanismo de atenção característico do professor.

Em relação ao DistilBERT aplicado a tarefas de menor complexidade, a exemplo do nível de Atividade, nota-se que o modelo aluno já possui capacidade suficiente para assimilar os padrões intrínsecos aos dados. Consequentemente, a introdução da função de perda associada à destilação pode ter atuado como um elemento de ruído ou de regularização excessiva, o que obstaculizou o ajuste fino ideal e justifica os valores negativos de $\Delta F1$. Contudo, observa-se uma inversão dessa lógica no nível de Etapa, dada a natureza da tarefa, caracterizada pela presença de múltiplas classes e fronteiras de decisão tênues. Nesse cenário, a KD desempenhou o papel de um regularizador benéfico ao orientar o modelo aluno em direção a uma generalização mais robusta, o que resultou nos ganhos evidenciados na Tabela 11.

A queda de desempenho da $textCNN$ no nível de Atividade sugere que a estrutura local das convoluções pode ter entrado em conflito com as distribuições de probabilidade globais geradas pelo modelo professor BERT, o que ocasionou um efeito de transferência negativa. Em contrapartida, nos níveis mais granulares, nos quais padrões locais e palavras-chave específicas são determinantes para a classificação, observou-se que esse conflito foi dissipado, permitindo que a arquitetura extraísse maior proveito do conhecimento destilado.

4.5 Estabilidade operacional e consistência das predições

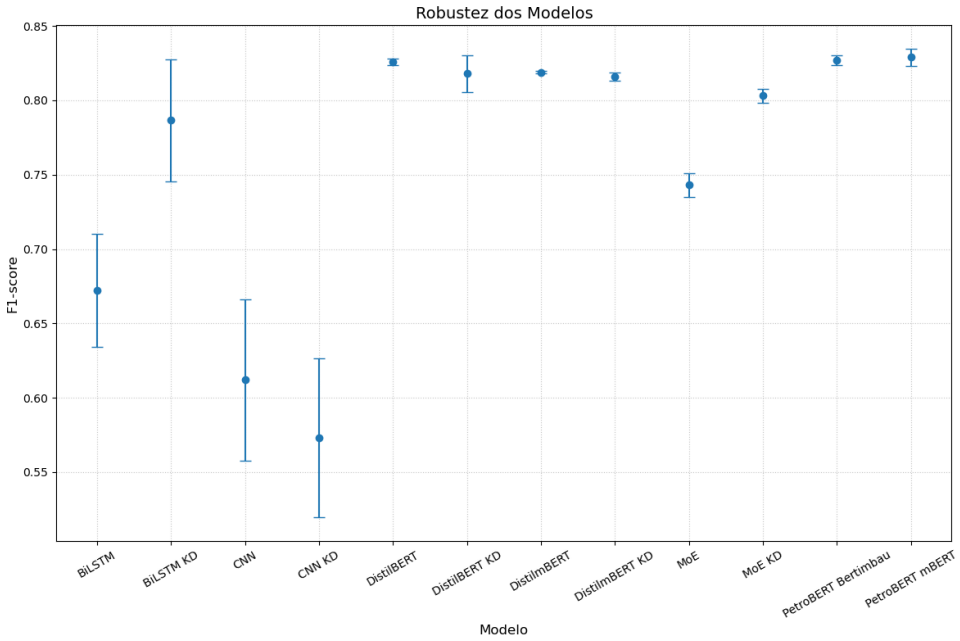
Nesta seção, a avaliação expande o escopo da métrica de desempenho médio, o F1-Score, para investigar a confiabilidade operacional das arquiteturas propostas. Compreende-se que, em um cenário industrial crítico, a consistência das predições é tão relevante quanto a precisão, uma vez que um modelo sujeito a alta variância implica um risco operacional elevado. Para quantificar essa característica, examina-se a dispersão dos resultados obtidos nas execuções independentes utilizando o CV. A análise fundamenta-se nos dados da Tabela 12, na inspeção dos boxplots de distribuição de F1 apresentados nas Figuras 21, 23 e 25, e também nos gráficos de barras de erro, ou *error bars* em inglês, os quais estão dispostos nas Figuras 20, 22 e 24, todos elaborados para os três níveis hierárquicos.

Tabela 12 – Análise de estabilidade do F1-Score médio para os três níveis hierárquicos.

Modelo	**Nível de Classificação**								
	Atividade			**Operação**			**Etapa**		
	μ	σ	**CV (%)**	μ	σ	**CV (%)**	μ	σ	**CV (%)**
CNN	0,6121	0,0543	8,9715	0,2518	0,0262	10,4223	0,1799	0,0121	6,7290
CNN_{KD}	0,5730	0,0535	9,3378	0,3044	0,0352	11,5654	0,2260	0,0233	10,3276
BiLSTM	0,6723	0,0380	5,6580	0,2480	0,0127	5,1166	0,2492	0,0150	5,9996
$BiLSTM_{KD}$	0,7865	0,0412	5,2378	0,3681	0,0199	5,4111	0,3616	0,0138	3,8091
MoE	0,7430	0,0079	1,0642	0,3322	0,0073	3,1238	0,3743	0,0044	1,2158
MoE_{KD}	0,8030	0,0045	0,5551	0,4930	0,0629	9,5401	0,4412	0,0049	0,8028
DistilBERT	0,8258	0,0024	0,2930	0,5563	0,0203	3,6559	0,4722	0,0047	1,0003
$DistilBERT_{KD}$	0,8180	0,0124	1,5108	0,5222	0,0267	5,1200	0,4954	0,0138	2,7903
DistilmBERT	0,8189	0,0010	0,1252	0,5382	0,0212	2,8792	0,4770	0,0057	1,6564
$DistilmBERT_{KD}$	0,8159	0,0025	0,3077	0,5316	0,0024	0,3312	0,5023	0,0024	0,6693

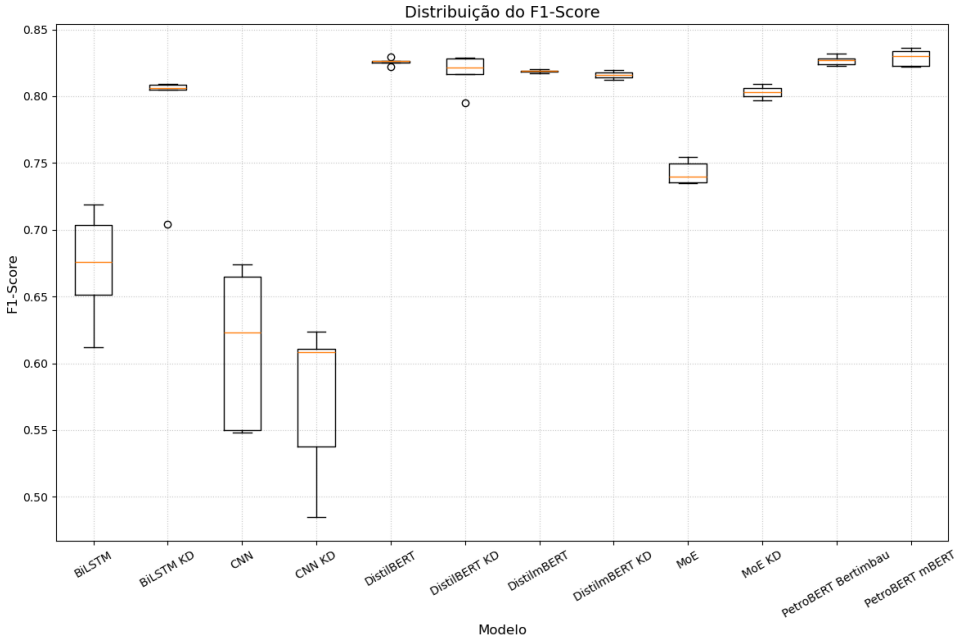
Fonte: Elaborada pelo autor.

Figura 20 – Intervalos de confiança do F1-Score indicando a robustez dos modelos para o nível de Atividade.



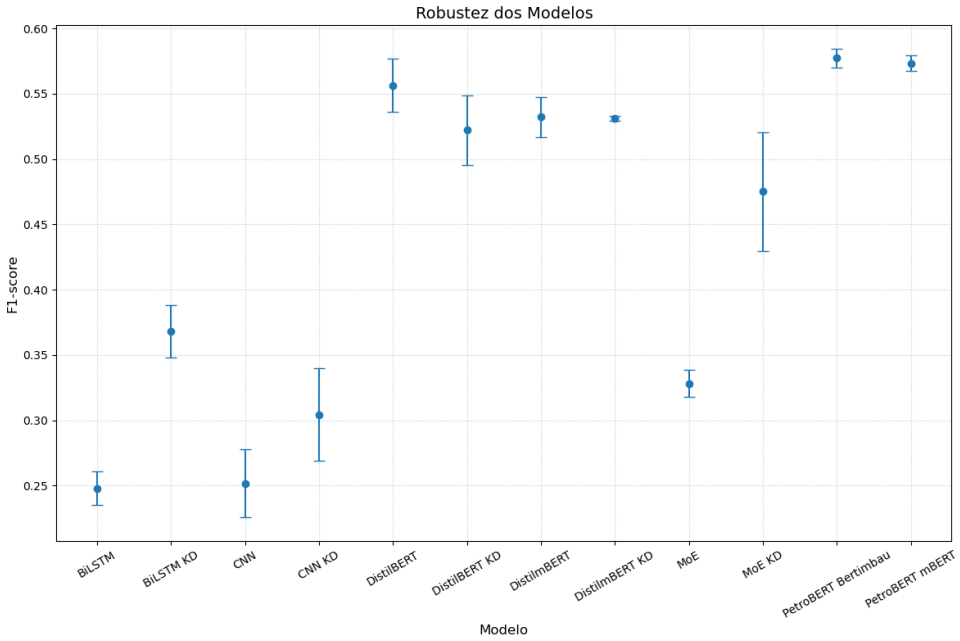
Fonte: Elaborada pelo autor.

Figura 21 – Boxplot da distribuição do F1-Score através das execuções independentes para o nível de Atividade.



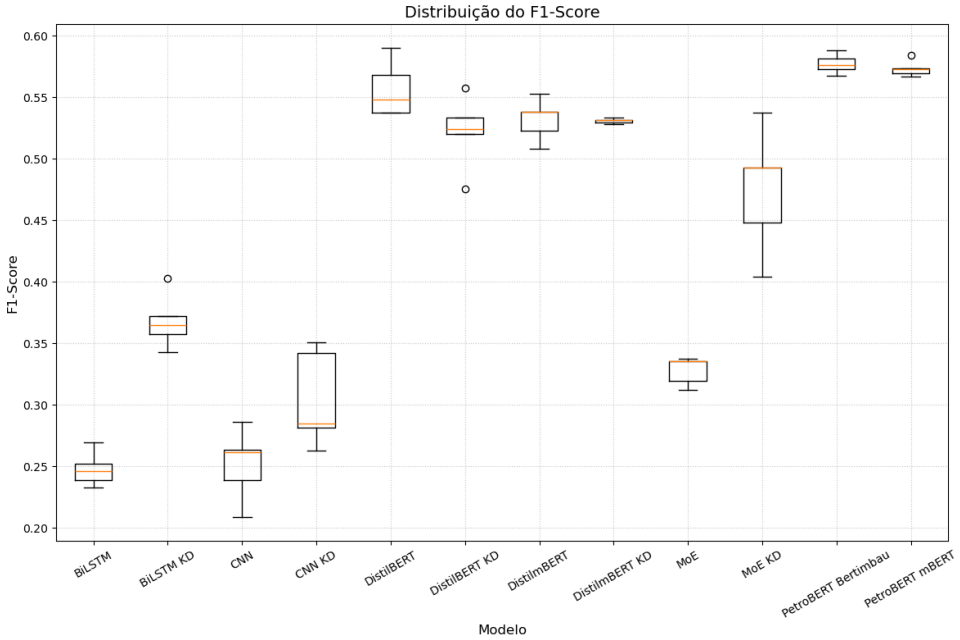
Fonte: Elaborada pelo autor.

Figura 22 – Intervalos de confiança do F1-Score indicando a robustez dos modelos para o nível de Operação.



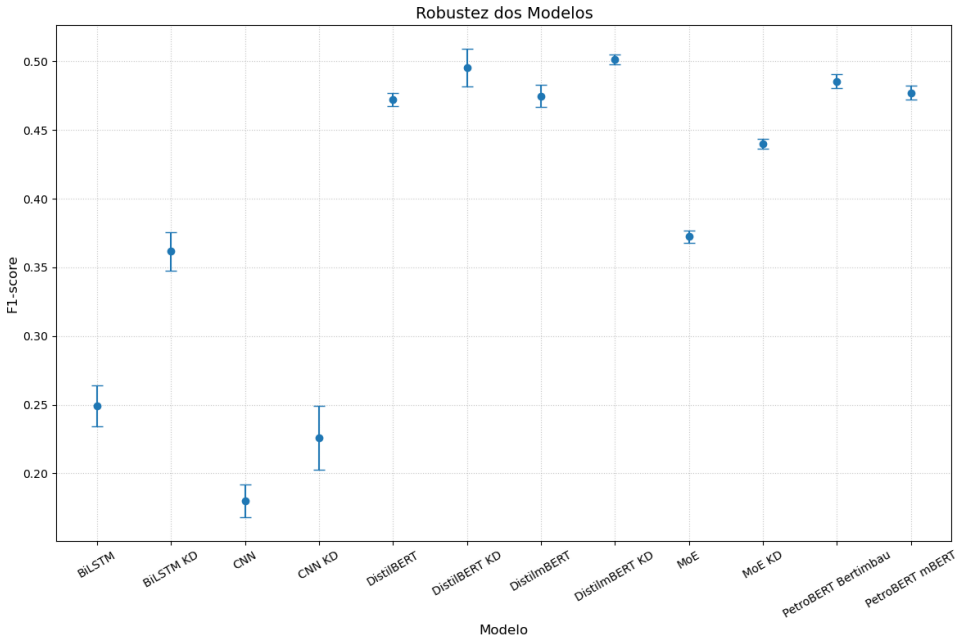
Fonte: Elaborada pelo autor.

Figura 23 – Boxplot da distribuição do F1-Score através das execuções independentes para o nível de Operação.



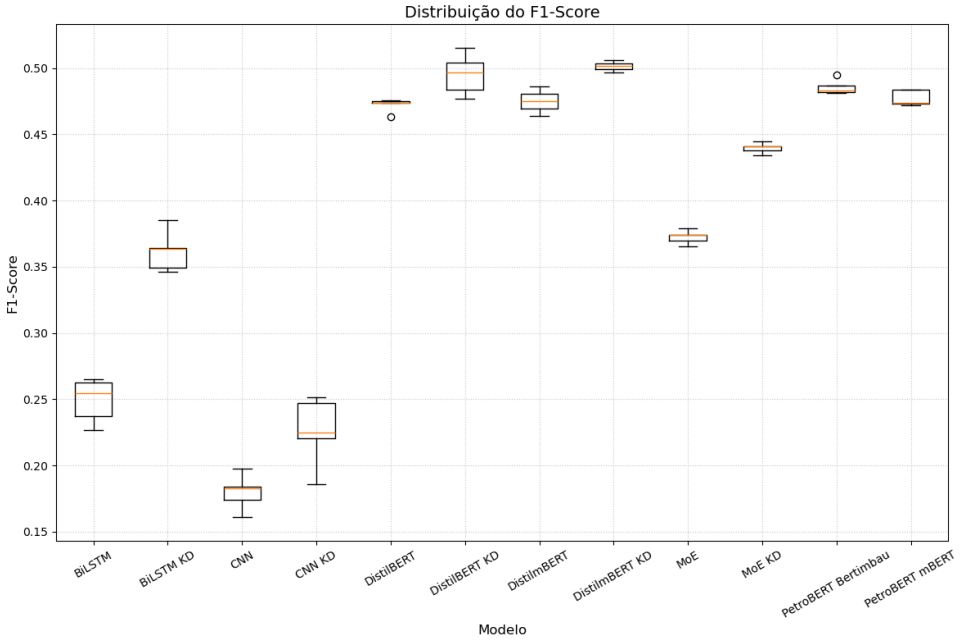
Fonte: Elaborada pelo autor.

Figura 24 – Intervalos de confiança do F1-Score indicando a robustez dos modelos para o nível de Etapa.



Fonte: Elaborada pelo autor.

Figura 25 – Boxplot da distribuição do F1-Score através das execuções independentes para o nível de Etapa.



Fonte: Elaborada pelo autor.

A Tabela 12 evidencia discrepâncias acentuadas na estabilidade observada entre as diferentes famílias de modelos. Os modelos baseados em mecanismos de atenção destacam-se pela robustez apresentada, como demonstra o DistilmBERT sem aplicação de KD, o qual

atingiu um CV notavelmente baixo de $0{,}1252\%$ no nível de Atividade e de $1{,}65\%$ no nível de Etapa. Por sua vez, a versão destilada DistilmBERT$_{\text{KD}}$ demonstrou consistência superior nos níveis de maior complexidade ao registrar um CV de apenas $0{,}33\%$ em Operação e $0{,}66\%$ em Etapa, valores significativamente inferiores aos observados nos demais modelos avaliados. Em contrapartida, as arquiteturas CNN apresentaram a maior volatilidade registrada no estudo, com o CV da CNN oscilando entre $6{,}7\%$ e $10{,}4\%$. Paradoxalmente, a aplicação da técnica de destilação na CNN$_{\text{KD}}$ intensificou a instabilidade em todos os cenários analisados, chegando a atingir um pico de $11{,}56\%$ no nível de Operação. Tal comportamento sugere que o modelo enfrenta dificuldades para convergir de forma consistente ao ser submetido ao processo de imitação do professor.

Já a BiLSTM sustenta uma estabilidade considerada moderada, com o CV situando-se em torno de 5%. Verifica-se que a técnica de destilação contribuiu para o aumento da robustez no nível de Etapa ao reduzir o coeficiente de $5{,}99\%$ para $3{,}80\%$. Simultaneamente, identifica-se uma anomalia no comportamento do modelo MoE$_{\text{KD}}$ no nível de Operação. Embora essa arquitetura apresente elevada estabilidade nos níveis de Atividade e Etapa, com índices de $0{,}55\%$ e $0{,}80\%$ respectivamente, o CV eleva-se abruptamente para $9{,}54\%$ em Operação. Tal disparidade sugere uma sensibilidade particular a esse conjunto de dados ou uma dificuldade de convergência dos especialistas nesse nível hierárquico intermediário.

Os *boxplots* corroboram visualmente os dados tabulares apresentados. Nos gráficos referentes às três hierarquias, percebe-se que as caixas representativas do intervalo interquartil dos modelos CNN e BiLSTM sem a aplicação de KD são significativamente mais alongadas e apresentam hastes extensas. Tal comportamento indica que, a depender da execução, o modelo pode exibir um desempenho que oscila entre excelente e medíocre, o que caracteriza baixa confiabilidade. Em contrapartida, os modelos DistilBERT, DistilmBERT e as variantes do PetroBERT exibem caixas extremamente compactas, as quais muitas vezes se assemelham a linhas horizontais. Isso denota que a convergência do modelo é robusta, independentemente da inicialização de seus parâmetros em uma determinada execução.

Outrossim, a presença de *outliers* na BiLSTM$_{\text{KD}}$, identificada nas figuras de Atividade e Operação, constitui um alerta. Embora a mediana se eleve com a utilização de KD, a existência de execuções com performance muito abaixo da média indica que a destilação, por si só, não elimina totalmente o risco de uma má inicialização em arquiteturas recorrentes. Já nos modelos PetroBERT, os *outliers* são raros ou situam-se muito próximos da mediana, o que sinaliza alta confiabilidade.

Observa-se que as barras de erro, que sintetizam a confiabilidade por meio do intervalo $\mu \pm \sigma$, apresentam-se quase imperceptíveis para os modelos BERT, assemelhando-se a pontos isolados. Identifica-se ainda um padrão visual relevante caracterizado pela redução da amplitude das barras de erro da BiLSTM em comparação à BiLSTMKD, notadamente na tarefa de Etapa, o que valida a hipótese de que o processo de destilação estabilizou essa arquitetura. Por fim, a

figura referente à robustez dos modelos em Operação destaca a anomalia do MoEKD, cuja barra de erro é perceptivelmente maior que a de seus pares baseados em BERT, comportamento que diverge do padrão de alta precisão sugerido por sua média de desempenho.

A estabilidade superior demonstrada pelas arquiteturas BERT, em especial o DistilBERT e o DistilmBERT, é atribuída aos mecanismos de pré-treinamento e de atenção. Visto que tais modelos partem de um estado de conhecimento linguístico prévio robusto, a etapa de ajuste fino tende a convergir para mínimos locais situados em regiões de grande proximidade na superfície de erro, independentemente do conjunto de dados empregado no treinamento. Em contrapartida, modelos treinados a partir do zero ou que fazem uso de *embeddings* estáticos, a exemplo da CNN e da BiLSTM, mostram-se altamente sensíveis à inicialização dos parâmetros em cada execução independente.

Nesse contexto, a influência da KD na estabilidade manifestou-se de maneira dual. No caso da arquitetura recorrente BiLSTM, a técnica atuou como um agente estabilizador ao reduzir a variância por meio do fornecimento de *soft targets*, os quais oferecem sinais de supervisão mais ricos e consistentes em comparação aos *hard labels*, orientando a descida do gradiente de forma mais segura durante o treinamento. Em contrapartida, para a rede CNN, a tentativa de replicar a complexa distribuição de probabilidade gerada pelo modelo BERT parece ter introduzido ruído no processo de otimização, o que exacerbou a instabilidade intrínseca dessa arquitetura.

4.6 Sustentabilidade e *trade-off* entre desempenho e custo

Esta seção final da análise de resultados dedica-se à dimensão da sustentabilidade computacional, em estrita consonância com os princípios de Green AI. O objetivo central consiste em quantificar o custo ambiental associado ao desempenho preditivo por meio da mensuração da relação de *trade-off* entre a precisão do modelo e sua pegada ecológica. A avaliação fundamenta-se nas Tabelas 13, 14 e 15, as quais detalham as reduções percentuais relativas representadas pelo símbolo Δ. Tais métricas referem-se à variação de desempenho no F1-Score, à emissão de gCO_2eq/kWh e ao consumo energético quando confrontadas com o modelo de referência PetroBERTBERTimbau. Adicionalmente, emprega-se a análise das Fronteiras de Pareto, ilustradas nas Figuras 26, 27, 28, 29, 30 e 31, com o intuito de correlacionar a eficácia aferida pelo F1-Score com o custo medido pelo tempo de inferência e pela emissão de gCO_2eq/kWh, o que viabiliza a identificação das soluções ótimas.

Tabela 13 – Redução percentual relativa de F1-Score, emissões de gCO_2eq/kWh e consumo energético em comparação ao *baseline* para o nível de Atividade.

Modelo	**Δ F1 (%)**	**Δ CO_2 (%)**	**Δ Energia (%)**
CNN	25,9670	98,0320	98,0320
CNN_{KD}	30,6960	98,1013	98,1013
BiLSTM	18,6904	98,6705	98,6705
$BiLSTM_{KD}$	4,8687	98,7660	98,7660
MoE	10,1336	88,6071	88,6071
MoE_{KD}	2,8742	88,5358	88,5358
DistilBERT	0,1179	46,8728	46,8728
$DistilBERT_{KD}$	1,0629	47,0378	47,0378
DistilmBERT	0,9552	46,9897	46,9897
$DistilmBERT_{KD}$	1,3134	46,9604	46,9604

Fonte: Elaborada pelo autor.

Tabela 14 – Redução percentual relativa de F1-Score, emissões de gCO_2eq/kWh e consumo energético em comparação ao *baseline* para o nível de Operação.

Modelo	**Δ F1 (%)**	**Δ CO_2 (%)**	**Δ Energia (%)**
CNN	56,3791	98,1144	98,1144
CNN_{KD}	47,2686	98,0742	98,0742
BiLSTM	57,0358	98,7490	98,7490
$BiLSTM_{KD}$	36,2332	98,7396	98,7396
MoE	43,1471	88,5495	88,5495
MoE_{KD}	17,6926	88,5089	88,5089
DistilBERT	3,6362	46,6045	46,6045
$DistilBERT_{KD}$	9,5435	46,6502	46,6502
DistilmBERT	7,8128	46,4361	46,4361
$DistilmBERT_{KD}$	8,0345	46,5543	46,5543

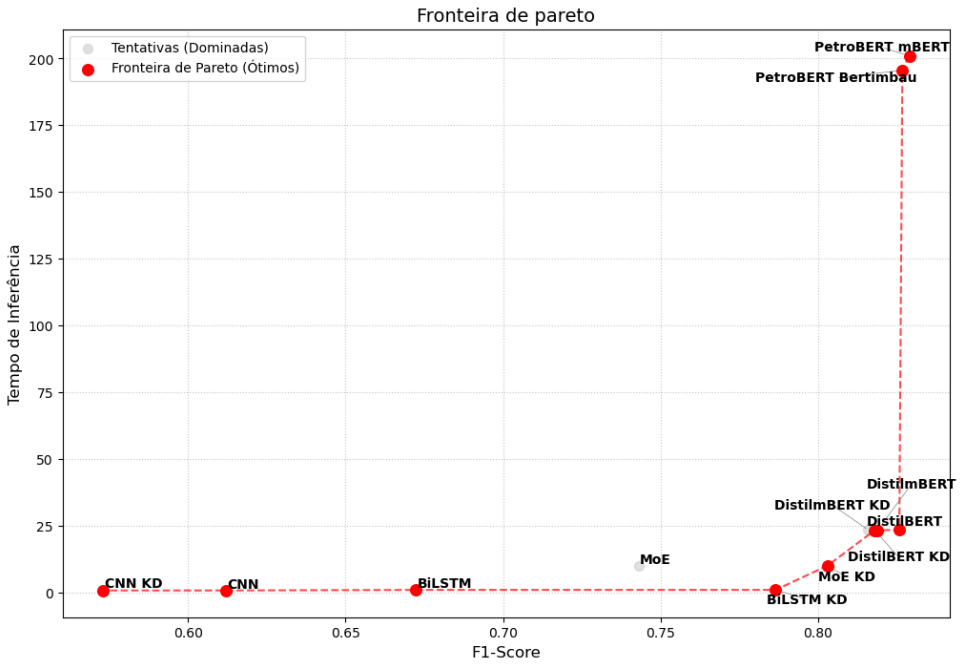
Fonte: Elaborada pelo autor.

Tabela 15 – Redução percentual relativa de F1-Score, emissões de gCO_2eq/kWh e consumo energético em comparação ao *baseline* para o nível de Etapa.

Modelo	**Δ F1 (%)**	**Δ CO_2 (%)**	**Δ Energia (%)**
CNN	62,9487	98,1306	98,1306
CNN_{KD}	53,4609	98,1209	98,1209
BiLSTM	48,6722	98,7523	98,7523
$BiLSTM_{KD}$	25,5270	98,7733	98,7733
MoE	23,2742	88,5663	88,5663
MoE_{KD}	9,4243	88,6482	88,6482
DistilBERT	2,7368	47,1396	47,1396
$DistilBERT_{KD}$	−2,0310	47,2101	47,2101
DistilmBERT	2,2118	47,0924	47,0924
$DistilmBERT_{KD}$	−3,2647	47,1254	47,1254

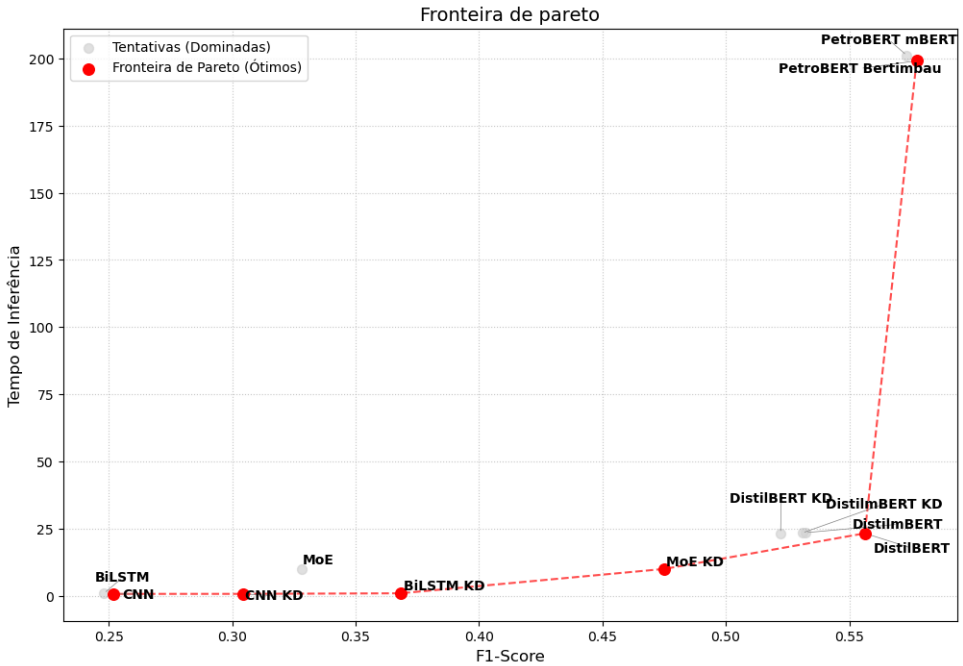
Fonte: Elaborada pelo autor.

Figura 26 – Fronteira de Pareto relacionando o Tempo de Inferência versus F1-Score para identificação de modelos ótimos em Atividade.



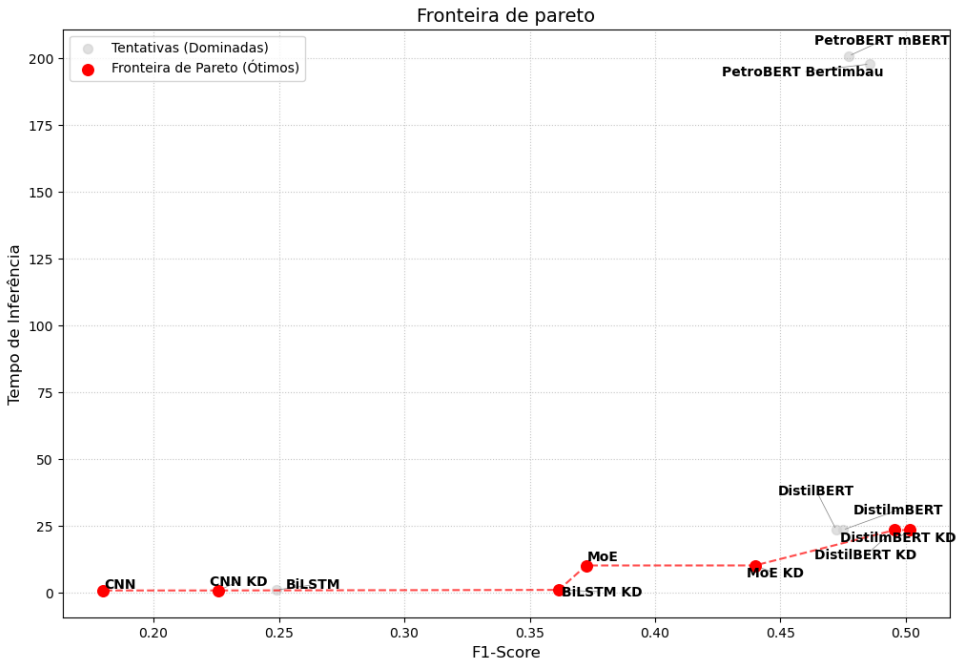
Fonte: Elaborada pelo autor.

Figura 27 – Fronteira de Pareto relacionando o Tempo de Inferência versus F1-Score para identificação de modelos ótimos em Operação.



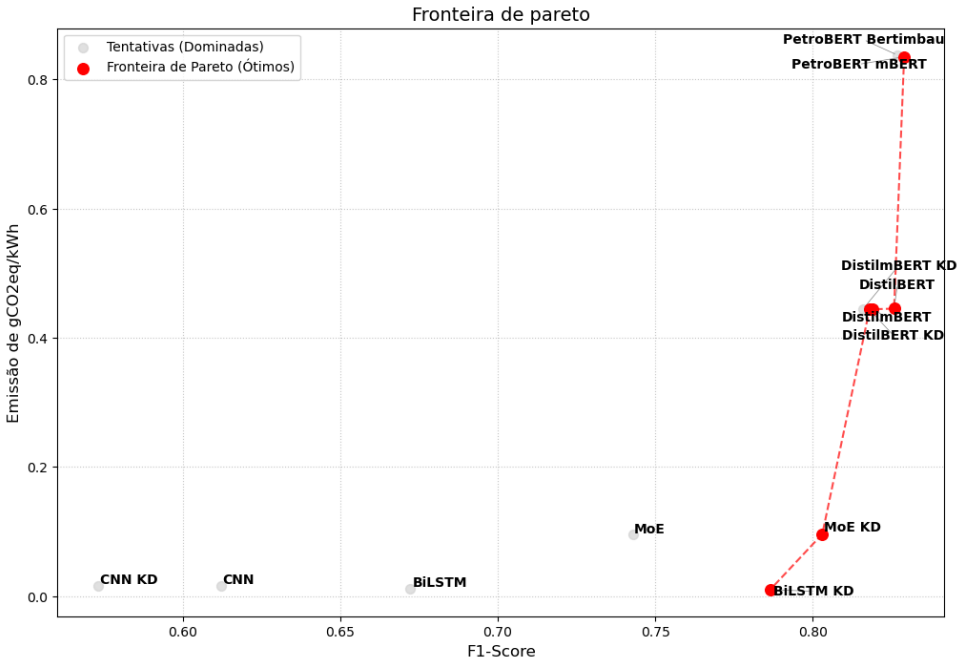
Fonte: Elaborada pelo autor.

Figura 28 – Fronteira de Pareto relacionando o Tempo de Inferência versus F1-Score para identificação de modelos ótimos em Etapa.



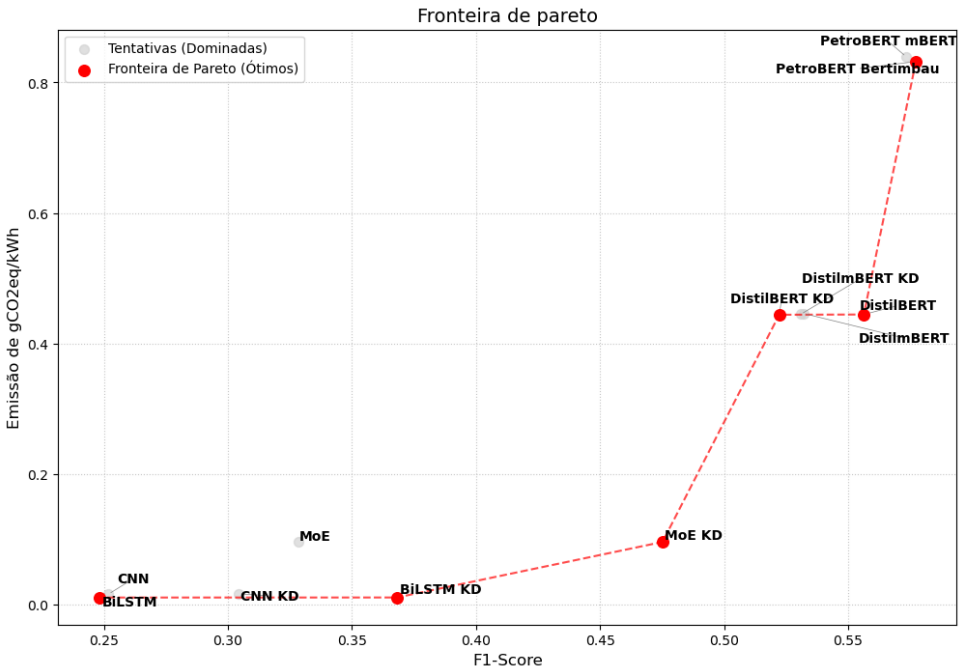
Fonte: Elaborada pelo autor.

Figura 29 – Fronteira de Pareto relacionando o a Emissão de gCO_2eq/kWh versus F1-Score para identificação de modelos ótimos em Atividade.



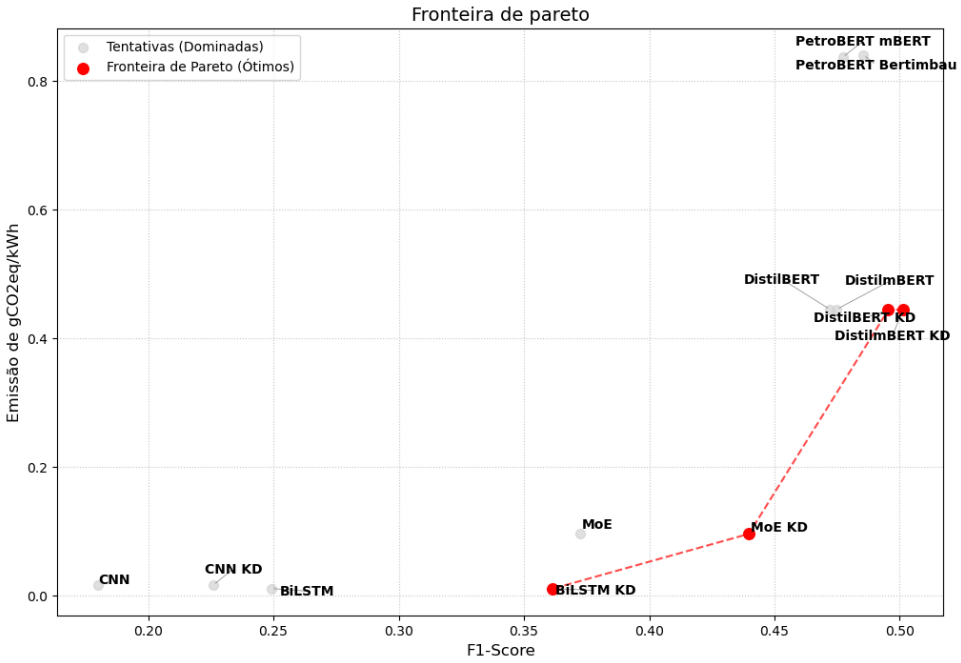
Fonte: Elaborada pelo autor.

Figura 30 – Fronteira de Pareto relacionando o a Emissão de gCO_2eq/kWh versus F1-Score para identificação de modelos ótimos em Operação.



Fonte: Elaborada pelo autor.

Figura 31 – Fronteira de Pareto relacionando o a Emissão de gCO_2eq/kWh versus F1-Score para identificação de modelos ótimos em Etapa.



Fonte: Elaborada pelo autor.

Os dados revelam que a eficiência energética e a redução de emissões constituem propriedades intrínsecas da arquitetura, mantendo-se constantes independentemente do nível

de classificação analisado. Em contrapartida, a perda de desempenho avaliada pelo F1-Score exibe variações drásticas, as quais se mostram diretamente condicionadas à complexidade da tarefa em questão.

As arquiteturas não baseadas em BERT destacam-se por oferecerem reduções substanciais de impacto ambiental. Observa-se que a CNN e a BiLSTM proporcionam uma economia de energia e uma diminuição de emissões de gCO_2eq/kWh situadas na faixa de $98{,}03\%$ a $98{,}77\%$ em todos os cenários analisados. Contudo, o comprometimento do desempenho preditivo revela-se proibitivo, haja vista que, no nível de Atividade, a CNN registra uma queda de $25{,}97\%$ no F1-Score, perda que se acentua para $62{,}95\%$ no nível de Etapa. Tal degradação inviabiliza a aplicação prática desses modelos, a despeito de sua notável eficiência ecológica. Em contrapartida, a família DistilBERT apresenta uma redução consistente de emissões e consumo energético próxima a 47%, com valores que oscilam entre $46{,}60\%$ e $47{,}21\%$. Diferentemente do observado nas arquiteturas mais leves, a penalidade de desempenho associada a esse ganho ambiental mostra-se mínima, uma vez que, no nível de Atividade, o DistilBERT sacrifica apenas $0{,}11\%$ de F1-Score para obter quase 47% de economia energética.

Um resultado digno de nota observa-se no tocante à classificação de Etapas. O modelo $\text{DistilBERT}_{\text{KD}}$ apresentou um $\Delta F1$ de $-2{,}03\%$, sendo imperativo destacar que, na métrica de perda de desempenho adotada, um valor negativo denota um ganho efetivo. Dessa forma, tal modelo não apenas reduziu o consumo de energia em $47{,}21\%$, como também superou a precisão do *baseline* em $2{,}03\%$. Comportamento análogo verifica-se no $\text{DistilmBERT}_{\text{KD}}$, o qual registrou um $\Delta F1$ de $-3{,}26\%$. O MoE posicionou-se como um meio-termo, oferecendo reduções de carbono na casa de 88%, mas com penalidades de F1 significativas em tarefas complexas (perda de $43{,}15\%$ em Operações), embora a versão KD tenha amortecido essa queda para $17{,}69\%$.

As figuras apresentadas ilustram as Fronteiras de Pareto, nas quais o eixo das abcissas (x) denota o F1-Score, objetivo de maximização, enquanto o eixo das ordenadas (y) representa a Emissão de gCO_2eq/kWh ou o Tempo de Inferência, métricas sujeitas à minimização. Em todos os gráficos, a fronteira exibe uma forma convexa acentuada que se assemelha visualmente a um "L"invertido ou a um "cotovelo". Tal configuração evidencia a existência de *trade-offs* não lineares, situação em que pequenos ganhos de desempenho na região superior direita, ocupada pelos modelos PetroBERT, acarretam aumentos exponenciais em emissões e tempo de processamento. Observa-se que os pontos vermelhos, indicativos da Fronteira Ótima, dominam consistentemente os pontos cinzas, referentes às tentativas dominadas. A análise visual permite constatar que as versões submetidas à destilação de conhecimento, a exemplo do $\text{DistilBERT}_{\text{KD}}$ e da $\text{BiLSTM}_{\text{KD}}$, tendem a deslocar os modelos para a direita em direção a um maior F1-Score sem elevar os custos representados no eixo y, o que os posiciona efetivamente na fronteira de eficiência.

No canto inferior esquerdo, caracterizado por baixa emissão e baixo F1-Score, encontram-

se aglomerados os modelos CNN e BiLSTM. Essas arquiteturas mostram-se extremamente rápidas e eficientes, posicionando-se próximas a zero no eixo y, porém distantes do desempenho ideal projetado no eixo x. No ponto de inflexão da curva, frequentemente denominado "joelho", situam-se os modelos DistilBERT e DistilmBERT, juntamente com suas versões destiladas. Tais modelos representam o ponto de equilíbrio do sistema, onde o desempenho aproxima-se significativamente do máximo obtido pelo PetroBERT, mas com uma redução substancial no custo representado no eixo y. A similaridade visual observada entre os gráficos de emissão de gCO_2eq/kWh e de tempo de inferência corrobora a existência de uma relação linear direta entre o tempo de uso de GPU ou TPU e a pegada de carbono gerada, o que confirma que a otimização de latência resulta diretamente em benefícios ambientais.

A análise integrada das tabelas e das Fronteiras de Pareto consolida a premissa de que os modelos de DL atuais, em especial aqueles fundamentados na arquitetura BERT, operam em uma zona de retornos decrescentes. A inclinação vertical da curva ao conectar o agrupamento do DistilBERT ao do PetroBERT evidencia o elevado custo marginal da precisão. Observa-se que, para obter ganhos fracionários de F1-Score, ou mesmo para registrar perda de desempenho no cenário específico de Etapa, o sistema incorre na duplicação das emissões de gCO_2eq/kWh e em um aumento de oito vezes no tempo de inferência. O PetroBERT demonstra-se, portanto, ambientalmente ineficiente para esta tarefa, pois situa-se na zona de saturação da curva, onde o investimento energético não se traduz em retorno proporcional de eficácia.

As arquiteturas baseadas em DistilBERT são identificadas como as soluções Pareto-ótimas preferenciais, visto que maximizam o desempenho antes que a curva de custo assuma um comportamento exponencial. A KD atua ao deslocar esse ponto de operação para a direita, o que resulta em um maior F1-Score sem elevar o custo, otimizando dessa forma a fronteira de eficiência do sistema. Em contrapartida, para cenários que exigem um compromisso intermediário, como dispositivos embarcados em campo (IoT) ou ambientes com restrições severas de bateria e energia, o MoE_{KD} ou a $BiLSTM_{KD}$ tornam-se as alternativas racionais. Essas arquiteturas sacrificam uma fração da precisão em troca de uma operação virtualmente isenta de custos computacionais significativos, alcançando reduções de energia de aproximadamente 88% e 99%, respectivamente.

Embora as CNNs e LSTMs dominem o extremo esquerdo da fronteira, setor caracterizado por um custo próximo a zero, tais arquiteturas falham no cumprimento do requisito funcional mínimo estabelecido. Elas representam, portanto, uma falsa economia, uma vez que o reduzido consumo de recursos é acompanhado por uma utilidade prática insuficiente. Esse desempenho posiciona tais modelos em uma região da fronteira que não satisfaz as exigências operacionais demandadas pela indústria de óleo e gás.

4.7 Considerações finais

A presente investigação buscou determinar a viabilidade de conciliar a elevada capacidade de generalização dos modelos de linguagem no estado da arte, aplicados à classificação de BDPs em poços de petróleo, com os rigorosos requisitos de eficiência computacional demandados por ambientes industriais. A análise integrada dos dados permite concluir que a compressão de modelos, operacionalizada especificamente por meio da arquitetura DistilBERT em associação com a técnica de KD, não representa apenas uma alternativa factível, mas a solução ótima para o problema abordado. A seguir, sintetizam-se as evidências que fundamentam tal conclusão.

Os experimentos revelaram um padrão consistente no qual a eficiência energética e a redução de emissões manifestam-se como propriedades intrínsecas da arquitetura, mantendo-se estáveis independentemente da granularidade da tarefa. Em contrapartida, o desempenho preditivo mensurado pelo F1-Score mostrou-se sensível à complexidade semântica dos dados. Identificou-se uma relação de compromisso não linear, visto que, enquanto arquiteturas clássicas como a CNN e a BiLSTM oferecem reduções de custo computacional superiores a 98%, elas falham na captura da semântica complexa dos BDPs. Tal limitação resulta em perdas de desempenho proibitivas, as quais atingem o patamar de $62{,}95\%$ no nível de Etapa. Por outro lado, as arquiteturas baseadas em Transformers destilados, representadas pelo DistilBERT, atingiram um ponto de equilíbrio ideal ao reduzirem o consumo de recursos em aproximadamente 47% com impacto marginal ou nulo na precisão, chegando inclusive a superar o modelo professor em tarefas de grão fino.

A hipótese central de que arquiteturas neurais de complexidade reduzida podem substituir modelos massivos sem prejuízo operacional foi corroborada. Contudo, a análise dos resultados impõe a observação das seguintes nuances específicas:

- Os dados confirmam de maneira inequívoca a hipótese referente ao ganho de eficiência. A substituição do modelo de referência PetroBERT pelo DistilBERT resultou em uma aceleração do tempo de inferência de cerca de nove vezes, o que corresponde a uma redução de aproximadamente 202 segundos para 23,3 segundos por lote. Essa otimização revela-se operacionalmente crítica, uma vez que permite a transição de um processamento em lote de longa duração para uma latência compatível com aplicações em tempo quase real. Ademais, a diminuição na contagem de operações de ponto flutuante, que caiu de 23,71 T para 11,86 T, valida a adequação desses modelos a arquiteturas de hardware com restrições de memória e processamento, o que confirma que a redução de parâmetros se traduz diretamente em agilidade operacional.

- A hipótese de manutenção de desempenho foi validada estatisticamente. No nível de Atividade, a aplicação do teste-t pareado demonstrou que o DistilBERT é estatisticamente equivalente ao estado da arte, apresentando um valor de $p > 0{,}05$. Resultados ainda mais expressivos foram observados na validação da hipótese em cenários de alta complexidade,

como na classificação de Etapa. Nessa tarefa, a técnica de KD mitigou a degradação preditiva a ponto de inverter a lógica de perda esperada, uma vez que o modelo aluno DistilmBERT$_{KD}$ superou o desempenho do professor. Tal fenômeno evidencia que a menor capacidade do modelo, quando devidamente orientada, atua como um regularizador eficaz contra o sobreajuste em dados ruidosos.

Os resultados obtidos corroboram a hipótese de alinhamento com os princípios de Green AI. A redução de aproximadamente 47% nas emissões de gCO_2eq/kWh evidencia que a adoção de modelos eficientes transcende a esfera puramente técnica, configurando-se também como um imperativo de responsabilidade ecológica. Adicionalmente, a simplicidade arquitetural inerente aos modelos destilados, caracterizada pela redução de 109 milhões para 66 milhões de parâmetros, favorece a integração em sistemas legados e em dispositivos de borda, no contexto de *Edge AI*. Tal característica assegura maior longevidade e escalabilidade à solução proposta quando comparada ao modelo monolítico utilizado como *baseline*.

O objetivo geral de desenvolver e validar arquiteturas eficientes voltadas à indústria de petróleo e gás foi plenamente atingido. As evidências experimentais, consolidadas nas análises das Fronteiras de Pareto, demonstram que as arquiteturas propostas ocupam a região de eficiência ótima, correspondente ao ponto de inflexão ou cotovelo da curva, onde se observa o equilíbrio entre precisão e custo. O estudo comprovou a viabilidade de mitigar o custo computacional intrínseco ao DL sem sacrificar a acurácia necessária para a tomada de decisão industrial, o que estabelece um arcabouço metodológico robusto e passível de transferência para outros domínios de NLP industrial.

No mais, a otimização via KD revelou-se crucial para modelos não baseados em Transformers, a exemplo da BiLSTM, ao elevar seu patamar de desempenho, além de atuar como um mecanismo de refinamento de robustez para Transformers em tarefas complexas. A comparação direta com o PetroBERT evidenciou que os modelos massivos operam em uma zona de retornos decrescentes, na qual o custo marginal para a obtenção de ganhos fracionários no F1-Score se torna injustificável. Ao igualarem o *baseline* em tarefas gerais e o superarem em cenários específicos, os modelos leves colocam em xeque a hegemonia das arquiteturas de grande porte para tarefas de classificação em domínio fechado.

A redução da variabilidade, caracterizada por um CV inferior a 1% para os Transformers destilados, atende aos requisitos de previsibilidade exigidos pelo setor industrial. A estabilidade demonstrada assegura a confiabilidade da automação na análise de BDPs, reduzindo a latência decisória em campo e permitindo a realocação de especialistas, anteriormente dedicados à triagem manual, para atividades de maior valor agregado. No âmbito acadêmico, o estudo oferece contribuições relevantes ao demonstrar empiricamente a teoria da lacuna de capacidade no processo de destilação e ao propor uma metodologia de avaliação que integra métricas de Green AI ao desempenho preditivo. Tais constatações apresentam elevado potencial de

disseminação em conferências voltadas à Inteligência Artificial aplicada e à sustentabilidade computacional.

Apesar do êxito geral alcançado, os dados apontaram limitações intrínsecas às arquiteturas puramente convolucionais, como a CNN, que falharam na captura de dependências de longo prazo, tornando-se inviáveis para o processamento de textos técnicos complexos a despeito da rapidez de execução. Adicionalmente, observou-se uma saturação da técnica de KD em tarefas simples, a exemplo da classificação de Atividade, para modelos que já demonstravam competência, como o DistilBERT. Nesse cenário, a transferência de conhecimento não gerou ganhos adicionais, o que indica a existência de um teto de performance para a arquitetura sob tais condições.

Em síntese, este estudo consolida a tese de que a definição de estado da arte na aplicação industrial de IA deve ser reorientada, deslocando-se da maximização isolada de métricas de acurácia para a otimização do equilíbrio entre precisão, latência e sustentabilidade. As arquiteturas destiladas, validadas tanto estatística quanto ambientalmente, emergem como a solução Pareto-ótima, o que fundamenta a transição para as conclusões gerais desta dissertação. Fica estabelecido, portanto, que a eficiência não atua como antagonista da eficácia, mas sim como uma aliada imprescindível para a computação moderna.

5 Conclusão

A presente dissertação abordou o desafio crítico da classificação automática de BDPs na indústria de petróleo e gás, cenário caracterizado pelo vasto volume de dados não estruturados e pela necessidade imperativa de suporte à decisão em tempo real. Embora os modelos de DL baseados em codificadores dos Transformers, como o PetroBERT, tenham estabelecido o estado da arte em termos de eficácia preditiva, sua aplicação prática enfrenta barreiras significativas relacionadas ao elevado custo computacional, à alta latência de inferência e ao consumo energético proibitivo para ambientes com restrições de *hardware*. A pesquisa ratificou a motivação, tanto industrial quanto científica, de transcender a busca exclusiva pela acurácia, posicionando o estudo na interseção entre o PLN aplicado e a Green AI. A investigação evidenciou que a dependência de arquiteturas massivas e superparametrizadas, embora eficazes, revela-se insustentável em termos operacionais. Tal cenário impõe a necessidade de soluções capazes de harmonizar a precisão semântica com a eficiência no consumo de recursos.

Para mitigar as limitações identificadas, o estudo adotou uma metodologia sistemática e comparativa, investigando um espectro diversificado de arquiteturas neurais eficientes, incluindo CNNs, BiLSTMs e modelos baseados em BERT compactos, como o DistilBERT, além de arquiteturas modulares como o MoE. A estratégia metodológica integrou técnicas avançadas de compressão e otimização, com ênfase na KD, utilizada para transferir a capacidade de generalização de um modelo professor robusto, representado pelo PetroBERT, para modelos alunos mais leves. O protocolo experimental rigoroso, validado nos três níveis hierárquicos de classificação, compreendendo Atividade, Operação e Etapa, assegurou que as conclusões sobre eficiência e desempenho não fossem artefatos de variações aleatórias, mas sim evidências sólidas do comportamento arquitetural.

Os resultados obtidos confirmam a primeira premissa da hipótese, demonstrando que as arquiteturas leves proporcionaram reduções drásticas no consumo de recursos. O modelo DistilBERT, por exemplo, alcançou uma aceleração de inferência de aproximadamente nove vezes em comparação ao *baseline*, ao reduzir o tempo de processamento de lotes de cerca de 200 s para 23 s, além de diminuir o consumo energético e as emissões de carbono na ordem de 47%. Modelos como a CNN e a BiLSTM ofereceram ganhos ainda mais expressivos, superiores a 98% em eficiência energética, embora apresentem compensações de desempenho distintas. Tais dados comprovam a viabilidade operacional dessas soluções em ambientes com restrições computacionais.

A hipótese de manutenção de desempenho foi confirmada, com destaque para as arquiteturas DistilBERT. O estudo evidenciou que a redução da complexidade não acarreta, necessariamente, uma degradação significativa da eficácia. No nível de Atividade, o DistilBERT

demonstrou equivalência estatística em relação ao PetroBERT. De maneira ainda mais relevante, no nível de maior complexidade, referente à classificação de Etapa, a aplicação de KD permitiu que os modelos compactos superassem o desempenho do modelo professor. Tal fenômeno inverte a lógica de perda esperada e comprova que a capacidade reduzida pode atuar como um regularizador eficaz contra o sobreajuste.

As soluções propostas alinham-se integralmente aos princípios de Green AI e sustentabilidade computacional. A análise de Fronteira de Pareto identificou os modelos destilados como soluções ótimas, uma vez que maximizam o retorno de desempenho por unidade de energia consumida. A redução significativa no número de parâmetros, que passou de 109 milhões para 66 milhões no caso do DistilBERT, facilita a manutenibilidade e a integração em sistemas legados, promovendo assim a longevidade tecnológica das soluções desenvolvidas. Dessa forma, considera-se que o objetivo geral de desenvolver e validar arquiteturas neurais de complexidade reduzida para a classificação de BDPs foi plenamente atingido. O estudo obteve êxito ao criar um conjunto de modelos capazes de mitigar o custo computacional e a latência sem sacrificar a precisão necessária para a aplicação industrial. A validação empírica robusta, fundamentada em testes estatísticos e métricas ambientais, conferiu solidez à conclusão de que é viável e desejável a substituição de modelos monolíticos por alternativas eficientes em cenários de produção.

A investigação permitiu constatar que as arquiteturas baseadas em *Transformers* destilados, notadamente o DistilBERT e o DistilmBERT, proporcionam o melhor equilíbrio global entre eficiência e desempenho no processamento de textos técnicos não estruturados. Ademais, a técnica de KD revelou-se fundamental, sobretudo para a otimização de modelos como a BiLSTM e o MoE, ao elevar significativamente seus patamares de desempenho em tarefas de alta complexidade. A análise comparativa demonstrou que os modelos convencionais, representados pelo PetroBERT, operam em uma zona de retornos decrescentes na qual o custo marginal necessário para a obtenção de ganhos fracionários de precisão se torna injustificável. As abordagens propostas alcançaram desempenho competitivo, atingindo a equivalência estatística em tarefas gerais e a superioridade em tarefas granulares, o que fundamenta a substituição dos modelos existentes na maioria das aplicações práticas.

Os resultados obtidos impulsionam a automação da classificação de BDPs ao viabilizar o processamento em tempo quase real, reduzindo consequentemente a latência decisória. A estabilidade operacional demonstrada pelos modelos destilados, cujos coeficientes de variação se mantiveram inferiores a 1%, atende aos rigorosos requisitos de confiabilidade e previsibilidade exigidos pela indústria, o que potencializa a adoção efetiva dessas soluções. O trabalho contribui para o estado da arte ao fornecer evidências empíricas acerca da lacuna de capacidade na destilação de conhecimento e ao validar um protocolo experimental que integra métricas de desempenho preditivo e impacto ambiental. Por fim, a discussão conduzida sob a ótica de Green AI estabelece um precedente relevante para futuras investigações que busquem conciliar a Inteligência Artificial avançada com a responsabilidade ecológica.

Torna-se necessário, contudo, reconhecer as limitações do estudo. As arquiteturas puramente convolucionais, ou CNNs, demonstraram incapacidade de capturar dependências de longo prazo em textos técnicos complexos, resultando em desempenho insatisfatório a despeito da alta eficiência. Além disso, observou-se uma saturação da técnica de KD em tarefas de menor complexidade, como no nível de Atividade, situação em que a transferência de conhecimento não gerou ganhos adicionais para modelos que já possuíam alta competência. Por fim, as restrições inerentes ao desbalanceamento das classes nos dados industriais impuseram tetos de desempenho em categorias sub-representadas.

Como desdobramentos naturais desta pesquisa, sugere-se:

- Investigar arquiteturas que transcendem a modelagem puramente sequencial, com destaque para as redes neurais em grafos, ou Graph Neural Network em inglês, capazes de capturar relações complexas entre equipamentos e operações. Sugere-se também a análise de Modelos de Espaço de Estados, como Mamba ou S4, que oferecem complexidade linear e inferência acelerada, bem como de arquiteturas híbridas neuro-simbólicas, as quais integram conhecimento de ontologias para ampliar a interpretabilidade.
- Expandir o escopo de otimização para além da destilação de conhecimento, avaliando o impacto de pruning, e da quantização consciente de treinamento, ou Quantization Aware Training (QAT) em inglês, para baixa precisão, como INT8 ou INT4. Recomenda-se ainda o emprego de NAS visando à descoberta de topologias otimizadas para *hardware* específico.
- Aumentar a robustez das conclusões mediante a ampliação do protocolo experimental para 15 ou mais execuções independentes. É recomendável a adoção de testes estatísticos globais, como Friedman e Nemenyi, ou de análises Bayesianas, além da implementação de validação cruzada temporal para respeitar a cronologia inerente aos dados industriais.
- Avaliar a resiliência dos modelos em cenários críticos, submetendo-os a testes de robustez adversarial, que incluem ruído e erros de digitação. Propõe-se também a verificação da capacidade de generalização Out-Of-Distribution (OOD) em dados provenientes de diferentes bacias sedimentares ou regimes de perfuração.

Referências

ADHIKARI, A.; RAM, A.; TANG, R.; HAMILTON, W. L.; LIN, J. Exploring the limits of simple learners in knowledge distillation for document classification with DocBERT. In: GELLA, S.; WELBL, J.; REI, M.; PETRONI, F.; LEWIS, P.; STRUBELL, E.; SEO, M.; HAJISHIRZI, H. (Ed.). **Proceedings of the 5th Workshop on Representation Learning for NLP**. Online: Association for Computational Linguistics, 2020. p. 72–77. Disponível em: <https://aclanthology.org/2020.repl4nlp-1.10/>. Acesso em: 05 jan. 2026.

BA, J.; CARUANA, R. Do deep nets really need to be deep? In: GHAHRAMANI, Z.; WELLING, M.; CORTES, C.; LAWRENCE, N.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2014. v. 27. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2014/file/ea8fcd92d59581717e06eb187f10666d-Paper.pdf>. Acesso em: 27 mar. 2023.

BROWN, T. B.; MANN, B.; RYDER, N.; SUBBIAH, M.; KAPLAN, J.; DHARIWAL, P.; NEELAKANTAN, A.; SHYAM, P.; SASTRY, G.; ASKELL, A.; AGARWAL, S.; HERBERT-VOSS, A.; KRUEGER, G.; HENIGHAN, T.; CHILD, R.; RAMESH, A.; ZIEGLER, D. M.; WU, J.; WINTER, C.; HESSE, C.; CHEN, M.; SIGLER, E.; LITWIN, M.; GRAY, S.; CHESS, B.; CLARK, J.; BERNER, C.; MCCANDLISH, S.; RADFORD, A.; SUTSKEVER, I.; AMODEI, D. Language models are few-shot learners. **arXiv preprint arXiv:2005.14165**, 2020. Disponível em: <https://doi.org/10.48550/arXiv.2005.14165>. Acesso em: 24 jun. 2023.

BUCILUǍ, C.; CARUANA, R.; NICULESCU-MIZIL, A. Model compression. In: **Anais de evento da 12th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2006. (KDD '06), p. 535–541. ISBN 1595933395. Disponível em: <https://doi.org/10.1145/1150402.1150464>. Acesso em: 13 mai. 2023.

CASELI, H. M.; NUNES, M. G. V. (Ed.). **Processamento de Linguagem Natural: Conceitos, Técnicas e Aplicações em Português**. 2. ed. BPLN, 2024. ISBN 978-65-00-95750-1. Disponível em: <https://brasileiraspln.com/livro-pln/2a-edicao/>. Acesso em: 28 jun. 2024.

CHENG, Y.; WANG, D.; ZHOU, P.; ZHANG, T. Model compression and acceleration for deep neural networks: The principles, progress, and challenges. **IEEE Signal Processing Magazine**, v. 35, n. 1, p. 126–136, jan. 2018. ISSN 1558-0792.

CHO, J.; HARIHARAN, B. On the efficacy of knowledge distillation. In: **2019 IEEE/CVF International Conference on Computer Vision (ICCV)**. Los Alamitos, CA, USA: IEEE Computer Society, 2019. p. 4793–4801. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/ICCV.2019.00489>. Acesso em: 21 mai. 2023.

CHOUDHARY, T.; MISHRA, V.; GOSWAMI, A.; SARANGAPANI, J. A comprehensive survey on model compression and acceleration. **Artificial Intelligence Review**, v. 53, n. 7, p. 5113–5155, out. 2020. ISSN 1573-7462. Disponível em: <https://doi.org/10.1007/s10462-020-09816-7>. Acesso em: 25 mar. 2023.

CINELLI, L. P.; de Oliveira, J. F.; de Pinho, V. M.; PASSOS, W. L.; PADILLA, R.; BRAZ, P. F.; GALVES, B.; DALVI, D. P.; LEWENFUS, G.; FERREIRA, J. O.; JI, A. Y.; de Oliveira,

F. L.; GONçALVES, C. J.; NETTO, S. L.; da Silva, E. A.; de Campos, M. L. Automatic event identification and extraction from daily drilling reports using an expert system and artificial intelligence. **Journal of Petroleum Science and Engineering**, v. 205, p. 108939, 2021. ISSN 0920-4105. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0920410521005982>. Acesso em: 11 mar. 2023.

COHEN, J. **Statistical power analysis for behavioral sciences**. 2nd. ed. Hillsdale, NY: Lawrence Erlbaum Associates, 1988. v. 1. 24-26 p. ISSN 09637214.

COURTY, B.; SCHMIDT, V.; LUCCIONI, S.; GOYAL-KAMAL; MARIONCOUTAREL; FELD, B.; LECOURT, J.; LIAMCONNELL; SABONI, A.; INIMAZ; SUPATOMIC; LéVAL, M.; BLANCHE, L.; CRUVEILLER, A.; OUMINASARA; ZHAO, F.; JOSHI, A.; BOGROFF, A.; LAVOREILLE, H. de; LASKARIS, N.; ABATI, E.; BLANK, D.; WANG, Z.; CATOVIC, A.; ALENCON, M.; STęCHłY, M.; BAUER, C.; ARAúJO, L. O. N. de; JPW; MINERVABOOKS. **mlco2/codecarbon: v2.4.1**. Zenodo, 2024. Disponível em: <https://doi.org/10.5281/zenodo.11171501>. Acesso em: 02 fev. 2026.

DEB, K. Multi-objective optimisation using evolutionary algorithms: An introduction. In: _____. **Multi-objective Evolutionary Optimisation for Product Design and Manufacturing**. London: Springer London, 2011. p. 3–34. ISBN 978-0-85729-652-8. Disponível em: <https://doi.org/10.1007/978-0-85729-652-8_1>. Acesso em: 02 fev. 2026.

DENG, L.; LI, G.; HAN, S.; SHI, L.; XIE, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. **Anais de evento da IEEE**, v. 108, n. 4, p. 485–532, abr. 2020. ISSN 1558-2256.

DEVLIN, J.; CHANG, M.-W.; LEE, K.; TOUTANOVA, K. Bert: Pre-training of deep bidirectional transformers for language understanding. **arXiv preprint arXiv:1810.04805**, 2019. Disponível em: <https://doi.org/10.48550/arXiv.1810.04805>. Acesso em: 24 jun. 2023.

FEDUS, W.; ZOPH, B.; SHAZEER, N. Switch transformers: Scaling to trillion parameter models with simple and efficient sparsity. **Journal of Machine Learning Research**, v. 23, n. 120, p. 1–39, 2022. Disponível em: <http://jmlr.org/papers/v23/21-0998.html>. Acesso em: 2 jan. 2026.

FEURER, M.; HUTTER, F. Hyperparameter optimization. In: HUTTER, F.; KOTTHOFF, L.; VANSCHOREN, J. (Ed.). **Automated Machine Learning: Methods, Systems, Challenges**. Cham: Springer International Publishing, 2019. p. 3–33. ISBN 978-3-030-05318-5. Disponível em: <https://doi.org/10.1007/978-3-030-05318-5_1>. Acesso em: 25 jun. 2023.

FILHO, J. A. W.; WILKENS, R.; IDIART, M.; VILLAVICENCIO, A. The brwac corpus: A new open resource for brazilian portuguese. In: **Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)**. [S.l.: s.n.], 2018.

GOLDBERG, D. E.; HOLLAND, J. H. Genetic algorithms and machine learning. **Machine Learning**, v. 3, n. 2, p. 95–99, out. 1988. ISSN 1573-0565. Disponível em: <https://doi.org/10.1023/A:1022602019183>. Acesso em: 24 jun. 2023.

GOU, J.; YU, B.; MAYBANK, S. J.; TAO, D. Knowledge distillation: A survey. **International Journal of Computer Vision**, v. 129, n. 6, p. 1789–1819, jun. 2021. ISSN 1573-1405. Disponível em: <https://doi.org/10.1007/s11263-021-01453-z>. Acesso em: 1 abr. 2023.

GUTMANN, M.; HYVäRINEN, A. Noise-contrastive estimation: A new estimation principle for unnormalized statistical models. In: TEH, Y. W.; TITTERINGTON, M. (Ed.). **Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics**. Chia Laguna Resort, Sardinia, Italy: PMLR, 2010. (Proceedings of Machine Learning Research, v. 9), p. 297–304. Disponível em: <https://proceedings.mlr.press/v9/gutmann10a.html>. Acesso em: 02 fev. 2026.

HE, K.; ZHANG, X.; REN, S.; SUN, J. Deep residual learning for image recognition. In: **2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Las Vegas, NV, USA: IEEE, 2016. p. 770–778.

HINTON, G.; VINYALS, O.; DEAN, J. Distilling the knowledge in a neural network. **arXiv preprint arXiv:1503.02531**, 2015. Disponível em: <https://doi.org/10.48550/arXiv.1503.02531>. Acesso em: 7 mai. 2023.

HOCHREITER, S.; SCHMIDHUBER, J. Long short-term memory. **Neural Comput.**, MIT Press, Cambridge, MA, USA, v. 9, n. 8, p. 1735–1780, nov. 1997. ISSN 0899-7667. Disponível em: <https://doi.org/10.1162/neco.1997.9.8.1735>. Acesso em: 24 jun. 2023.

HOFFIMANN, J.; MAO, Y.; WESLEY, A.; TAYLOR, A. Sequence mining and pattern analysis in drilling reports with deep natural language processing. In: **Anais de evento da SPE Annual Technical Conference and Exhibition**. OnePetro, 2018. (SPE Annual Technical Conference and Exhibition, Day 3 Wed, September 26, 2018). D031S033R004. Disponível em: <https://doi.org/10.2118/191505-MS>. Acesso em: 11 mar. 2023.

HOLM, S. A simple sequentially rejective multiple test procedure. **Scandinavian Journal of Statistics**, v. 6, p. 65–70, 1979. Disponível em: <https://www.ime.usp.br/~abe/lista/pdf4R8xPVzCnX.pdf>. Acesso em: 02 fev. 2026.

JOULIN, A.; GRAVE, E.; BOJANOWSKI, P.; MIKOLOV, T. Bag of tricks for efficient text classification. In: **Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers**. Valencia, Spain: Association for Computational Linguistics, 2017. p. 427–431. Disponível em: <https://aclanthology.org/E17-2068>. Acesso em: 24 jun. 2023.

KIM, Y. Convolutional neural networks for sentence classification. In: MOSCHITTI, A.; PANG, B.; DAELEMANS, W. (Ed.). **Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)**. Doha, Qatar: Association for Computational Linguistics, 2014. p. 1746–1751. Disponível em: <https://aclanthology.org/D14-1181/>. Acesso em: 01 fev. 2026.

KOZA, J. **Genetic Programming: On the Programming of Computers by Means of Natural Selection**. Stanford, CA, USA: Bradford, 1992. (A Bradford book). ISBN 9780262111706.

KRICHEN, M. Convolutional neural networks: A survey. **Computers**, v. 12, n. 8, 2023. ISSN 2073-431X. Disponível em: <https://www.mdpi.com/2073-431X/12/8/151>. Acesso em: 13 jan. 2026.

KRIZHEVSKY, A.; SUTSKEVER, I.; HINTON, G. E. Imagenet classification with deep convolutional neural networks. In: PEREIRA, F.; BURGES, C.; BOTTOU, L.; WEINBERGER, K. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc.,

2012. v. 25. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2012/file/c399862d3b9d6b76c8436e924a68c45b-Paper.pdf>. Acesso em: 25 jun. 2023.

LAFFERTY, J. D.; MCCALLUM, A.; PEREIRA, F. C. N. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: **Proceedings of the Eighteenth International Conference on Machine Learning**. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2001. (ICML '01), p. 282–289. ISBN 1558607781.

LECUN, Y.; BENGIO, Y. Convolutional networks for images, speech, and time series. In: _____. **The Handbook of Brain Theory and Neural Networks**. Cambridge, MA, USA: MIT Press, 1998. p. 255–258. ISBN 0262511029.

LECUN, Y.; DENKER, J.; SOLLA, S. Optimal brain damage. In: TOURETZKY, D. (Ed.). **Advances in Neural Information Processing Systems**. Morgan-Kaufmann, 1989. v. 2. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/1989/file/6c9882bbac1c7093bd25041881277658-Paper.pdf>. Acesso em: 24 jun. 2023.

LOSHCHILOV, I.; HUTTER, F. Decoupled weight decay regularization. In: **International Conference on Learning Representations**. [s.n.], 2019. Disponível em: <https://openreview.net/forum?id=Bkg6RiCqY7>. Acesso em: 30 jul. 2024.

LU, Y.; LIU, L.; NIE, R. Mixture-of-experts based llm model for financial text classification. In: **Proceedings of the 2024 5th International Conference on Computer Science and Management Technology**. New York, NY, USA: Association for Computing Machinery, 2025. (ICCSMT '24), p. 483–486. ISBN 9798400709999. Disponível em: <https://doi.org/10.1145/3708036.3708118>. Acesso em: 02 jan. 2026.

MA, Z.; VAJARGAH, A. K.; LEE, H.; KANSAO, R.; DARABI, H.; CASTINEIRA, D. Applications of machine learning and data mining in speedwise® drilling analytics: A case study. In: **Anais de evento da Abu Dhabi International Petroleum Exhibition & Conference**. OnePetro, 2018. (Abu Dhabi International Petroleum Exhibition and Conference, Day 2 Tue, November 13, 2018). D022S147R001. Disponível em: <https://doi.org/10.2118/193224-MS>. Acesso em: 19 mar. 2023.

MENGHANI, G. Efficient deep learning: A survey on making deep learning models smaller, faster, and better. **arXiv preprint arXiv:2106.08962**, 2021. Disponível em: <https://doi.org/10.48550/arXiv.2106.08962>. Acesso em: 1 abr. 2023.

MIKOLOV, T.; DEORAS, A.; POVEY, D.; BURGET, L.; ČERNOCKý, J. Strategies for training large scale neural network language models. In: **2011 IEEE Workshop on Automatic Speech Recognition & Understanding**. [S.l.: s.n.], 2011. p. 196–201.

MINAEE, S.; KALCHBRENNER, N.; CAMBRIA, E.; NIKZAD, N.; CHENAGHLU, M.; GAO, J. Deep learning–based text classification: A comprehensive review. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 54, n. 3, abr. 2021. ISSN 0360-0300. Disponível em: <https://doi.org/10.1145/3439726>. Acesso em: 29 jul. 2024.

MISHRA, R.; GUPTA, H. P.; DUTTA, T. A survey on deep neural network compression: Challenges, overview, and solutions. **arXiv preprint arXiv:2010.03954**, 2020. Disponível em: <https://doi.org/10.48550/arXiv.2010.03954>. Acesso em: 1 abr. 2023.

MU, S.; LIN, S. A comprehensive survey of mixture-of-experts: Algorithms, theory, and applications. **arXiv preprint arXiv:2503.07137**, 2025. Disponível em: <https://doi.org/10.48550/arXiv.2503.07137>. Acesso em: 22 dez. 2025.

PENG, B.; CHERSONI, E.; HSU, Y.-Y.; HUANG, C.-R. Is domain adaptation worth your investment? comparing BERT and FinBERT on financial tasks. In: HAHN, U.; HOSTE, V.; STENT, A. (Ed.). **Proceedings of the Third Workshop on Economics and Natural Language Processing**. Punta Cana, Dominican Republic: Association for Computational Linguistics, 2021. p. 37–44. Disponível em: <https://aclanthology.org/2021.econlp-1.5/>. Acesso em: 05 jan. 2026.

RIBEIRO, L. C.; AFONSO, L. C.; COLOMBO, D.; GUILHERME, I. R.; PAPA, J. P. Evolving neural conditional random fields for drilling report classification. **Journal of Petroleum Science and Engineering**, v. 187, p. 106846, 2020. ISSN 0920-4105. Disponível em: <https://www.sciencedirect.com/science/article/pii/S092041051931263X>. Acesso em: 10 mar. 2023.

RODRIGUES, R. B. M.; PRIVATTO, P. I. M.; SOUSA, G. J. de; MURARI, R. P.; AFONSO, L. C. S.; PAPA, J. P.; PEDRONETTE, D. C. G.; GUILHERME, I. R.; PERROUT, S. R.; RIENTE, A. F. Petrobert: A domain adaptation language model for oil and gas applications in portuguese. In: PINHEIRO, V.; GAMALLO, P.; AMARO, R.; SCARTON, C.; BATISTA, F.; SILVA, D.; MAGRO, C.; PINTO, H. (Ed.). **Computational Processing of the Portuguese Language**. Cham: Springer International Publishing, 2022. p. 101–109. ISBN 978-3-030-98305-5.

RUSSAKOVSKY, O.; DENG, J.; SU, H.; KRAUSE, J.; SATHEESH, S.; MA, S.; HUANG, Z.; KARPATHY, A.; KHOSLA, A.; BERNSTEIN, M.; BERG, A. C.; FEI-FEI, L. ImageNet Large Scale Visual Recognition Challenge. **International Journal of Computer Vision (IJCV)**, v. 115, n. 3, p. 211–252, 2015.

SANH, V.; DEBUT, L.; CHAUMOND, J.; WOLF, T. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. **arXiv preprint arXiv:1910.01108**, 2020. Disponível em: <https://doi.org/10.48550/arXiv.1910.01108>. Acesso em: 23 abr. 2023.

SCHWARTZ, R.; DODGE, J.; SMITH, N. A.; ETZIONI, O. Green ai. **Commun. ACM**, Association for Computing Machinery, New York, NY, USA, v. 63, n. 12, p. 54–63, nov. 2020. ISSN 0001-0782. Disponível em: <https://doi.org/10.1145/3381831>. Acesso em: 25 jul. 2024.

SHAPIRO, S. S.; WILK, M. B. An analysis of variance test for normality (complete samples)†. **Biometrika**, v. 52, n. 3-4, p. 591–611, 12 1965. ISSN 0006-3444. Disponível em: <https://doi.org/10.1093/biomet/52.3-4.591>. Acesso em: 02 fev. 2026.

SILVA, M.; OLIVEIRA, G.; COSTA, L.; PAPPA, G. Evaluating domain-adapted language models for governmental text classification tasks in portuguese. In: **Anais do XXXIX Simpósio Brasileiro de Bancos de Dados**. Porto Alegre, RS, Brasil: SBC, 2024. p. 247–259. ISSN 2763-8979. Disponível em: <https://sol.sbc.org.br/index.php/sbbd/article/view/30697>. Acesso em: 05 jan. 2026.

SIMONYAN, K.; ZISSERMAN, A. Very deep convolutional networks for large-scale image recognition. **arXiv preprint arXiv:1409.1556**, 2015. Disponível em: <https://doi.org/10.48550/arXiv.1409.1556>. Acesso em: 25 jun. 2023.

SOUSA, G. J.; PEDRONETTE, D. C. G.; BALDASSIN, A.; PRIVATTO, P. I. M.; GASETA, M.; GUILHERME, I. R.; COLOMBO, D.; AFONSO, L. C. S.; PAPA, J. P. Pattern analysis in drilling reports using optimum-path forest. In: **2018 International Joint Conference on Neural Networks (IJCNN)**. Rio de Janeiro, Brasil: IEEE, 2018. p. 1–8. ISSN 2161-4407.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. BERTimbau: pretrained BERT models for Brazilian Portuguese. In: **9th Brazilian Conference on Intelligent Systems, BRACIS**. Rio Grande do Sul, Brazil: Springer, 2020.

SOUZA, F.; NOGUEIRA, R.; LOTUFO, R. Portuguese named entity recognition using bert-crf. **arXiv preprint arXiv:1909.10649**, 2020. Disponível em: <https://doi.org/10.48550/arXiv.1909.10649>. Acesso em: 25 jun. 2023.

SPEER, R. **ftfy: fixes text for you**. 2019. Zenodo. Version 5.5. Disponível em: <https://doi.org/10.5281/zenodo.2591652>. Acesso em: 28 jul.2024.

SUN, S.; CHENG, Y.; GAN, Z.; LIU, J. Patient knowledge distillation for bert model compression. **arXiv preprint arXiv:1908.09355**, 2019. Disponível em: <https://doi.org/10.48550/arXiv.1908.09355>. Acesso em: 27 mai. 2023.

SZEGEDY, C.; LIU, W.; JIA, Y.; SERMANET, P.; REED, S.; ANGUELOV, D.; ERHAN, D.; VANHOUCKE, V.; RABINOVICH, A. Going deeper with convolutions. In: **2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)**. Los Alamitos, CA, USA: IEEE Computer Society, 2015. p. 1–9. ISSN 1063-6919. Disponível em: <https://doi.ieeecomputersociety.org/10.1109/CVPR.2015.7298594>. Acesso em: 24 jun. 2023.

TABBAKH, A.; AMIN, L. A.; ISLAM, M.; MAHMUD, G. M. I.; CHOWDHURY, I. K.; MUKTA, M. S. H. Towards sustainable ai: a comprehensive framework for green ai. **Discover Sustainability**, v. 5, n. 1, p. 408, nov. 2024. ISSN 2662-9984. Disponível em: <https://doi.org/10.1007/s43621-024-00641-4>. Acesso em: 07 jan. 2026.

TANG, R.; LU, Y.; LIU, L.; MOU, L.; VECHTOMOVA, O.; LIN, J. Distilling task-specific knowledge from bert into simple neural networks. **arXiv preprint arXiv:1903.12136**, 2019. Disponível em: <https://doi.org/10.48550/arXiv.1903.12136>. Acesso em: 28 mai. 2023.

VASWANI, A.; SHAZEER, N.; PARMAR, N.; USZKOREIT, J.; JONES, L.; GOMEZ, A. N.; KAISER, L. u.; POLOSUKHIN, I. Attention is all you need. In: GUYON, I.; LUXBURG, U. V.; BENGIO, S.; WALLACH, H.; FERGUS, R.; VISHWANATHAN, S.; GARNETT, R. (Ed.). **Advances in Neural Information Processing Systems**. Curran Associates, Inc., 2017. v. 30. Disponível em: <https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf>. Acesso em: 29 jul. 2024.

ZAGORUYKO, S.; KOMODAKIS, N. Paying more attention to attention: Improving the performance of convolutional neural networks via attention transfer. **arXiv preprint arXiv:1612.03928**, 2017. Disponível em: <https://doi.org/10.48550/arXiv.1612.03928>. Acesso em: 24 jun. 2023.

ZHANG, A.; LIPTON, Z. C.; LI, M.; SMOLA, A. J. **Dive into Deep Learning**. Cambridge University Press, 2023. Disponível em: <https://D2L.ai>. Acesso em: 29 jul. 2024.

ZHANG, C.; YANG, Y.; LIU, J.; WANG, J.; XIAN, Y.; WANG, B.; SONG, D. Lifting the curse of capacity gap in distilling language models. In: ROGERS, A.; BOYD-GRABER, J.; OKAZAKI, N. (Ed.). **Proceedings of the 61st Annual Meeting of the Association for**

Computational Linguistics (Volume 1: Long Papers). Toronto, Canada: Association for Computational Linguistics, 2023. p. 4535–4553. Disponível em: <https://aclanthology.org/2023.acl-long.249/>. Acesso em: 01 fev. 2026.

ZHAO, J.; ZHAO, Z.; SHI, L.; KUANG, Z.; LIU, Y. Collaborative mixture-of-experts model for multi-domain fake news detection. **Electronics**, v. 12, n. 16, 2023. ISSN 2079-9292. Disponível em: <https://www.mdpi.com/2079-9292/12/16/3440>. Acesso em: 02 jan. 2026.

ZUO, S.; ZHANG, Q.; LIANG, C.; HE, P.; ZHAO, T.; CHEN, W. MoEBERT: from BERT to mixture-of-experts via importance-guided adaptation. In: CARPUAT, M.; MARNEFFE, M.-C. de; RUIZ, I. V. M. (Ed.). **Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies**. Seattle, United States: Association for Computational Linguistics, 2022. p. 1610–1623. Disponível em: <https://aclanthology.org/2022.naacl-main.116/>. Acesso em: 02 jan. 2026.