

UNIVERSIDADE FEDERAL FLUMINENSE

NAOKI YOKOYAMA

***HYPERDIMENSIONAL COMPUTING PARA CLASSIFICAÇÃO DE
FAKE NEWS***

NITERÓI

2026

NAOKI YOKOYAMA

***HYPERDIMENSIONAL COMPUTING PARA CLASSIFICAÇÃO DE
FAKE NEWS***

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre em Computação. Área de Concentração: Ciência da Computação.

Orientador:

Prof. Dr. Leandro Santiago de Araújo

NITERÓI

2026

Ficha catalográfica automática - SDC/BCPV
Gerada com informações fornecidas pelo autor

Y54h Yokoyama, Naoki
Hyperdimensional Computing para Classificação de Fake News
/ Naoki Yokoyama. - 2026.
102 f.

Orientador: Leandro Santiago de Araújo.
Dissertação (mestrado)-Universidade Federal Fluminense,
Instituto de Computação, Niterói, 2026.

1. Computação hiperdimensional. 2. Notícias falsas. 3.
Processamento de linguagem natural. 4. Aprendizado de
máquina. 5. Produção intelectual. I. Araújo, Leandro
Santiago de, orientador. II. Universidade Federal Fluminense.
Instituto de Computação. III. Título.

CDD - XXX

NAOKI YOKOYAMA

***HYPERDIMENSIONAL COMPUTING PARA CLASSIFICAÇÃO DE FAKE
NEWS***

Dissertação apresentada ao Programa de Pós-Graduação em Computação da Universidade Federal Fluminense, como requisito parcial para obtenção do Grau de Mestre em Computação. Área de Concentração: Ciência da Computação.

Aprovada em 14 de setembro de 2026.

BANCA EXAMINADORA

Prof. Dr. Leandro Santiago de Araújo – Orientador
UFF

Profª. Dra. Aline Marins Paes Carvalho
UFF

Prof. Dr. Leopoldo André Dutra Lusquino Filho
UNESP

Niterói

2026

AGRADECIMENTOS

À minha esposa, Rachel de Almeida, agradeço por todo o amor, a paciência e o apoio incondicional ao longo de cada etapa desta caminhada. Sua presença constante, seu incentivo nos momentos difíceis e sua compreensão nas muitas horas dedicadas a este trabalho foram essenciais para que eu chegasse até aqui. Sem você, nada disso seria possível.

À minha filha, Michaela Yokoyama, que é a minha maior alegria e a razão do meu esforço diário. Que este trabalho sirva um dia como exemplo de que a dedicação e a persistência valem a pena.

Um agradecimento especial à minha cunhada, Sabine de Almeida Azevedo, pelo apoio generoso e pela presença sempre solícita, que de tantas formas tornaram esta jornada mais leve e possível.

De modo muito especial, agradeço ao meu grande orientador, Prof. Dr. Leandro Santiago, pela atenção, paciência e nobreza no trato, e pelo empenho constante em fazer com que este trabalho se concretizasse. Sua orientação criteriosa e generosa foi determinante em cada página desta dissertação; levo comigo, além do conhecimento, o exemplo de dedicação e de rigor acadêmico.

A todos que, direta ou indiretamente, contribuíram para a realização desta pesquisa, o meu sincero muito obrigado.

RESUMO

A disseminação massiva de desinformação em ambientes digitais — fenômeno que já afetava processos eleitorais e o debate público antes de 2020 e que se intensificou durante a pandemia de COVID-19 — evidenciou a necessidade de sistemas automatizados de detecção de *fake news* capazes de operar em larga escala. Entretanto, o estado da arte da área tem sido dominado por arquiteturas baseadas em *Deep Learning*, *Transformers* e *Large Language Models* (LLMs), que apresentam elevado custo computacional, alto consumo energético e limitada aplicabilidade em cenários de borda. Diante desse contexto, esta dissertação investiga a viabilidade da Computação Hiperdimensional (*Hyperdimensional Computing* – HDC) como paradigma alternativo, fundamentado em vetores de alta dimensão e operações algébricas leves, sob a perspectiva do *trade-off* entre acurácia preditiva e eficiência energética. A pesquisa adota abordagem quantitativa-experimental e avalia o desempenho de cinco variantes de HDC (*Vanilla* HDC, *AdaptHD*, *OnlineHD*, *NeuralHD* e *DistHD*) frente a modelos de referência de Aprendizado de Máquina clássico (*Logistic Regression*, SVM, *Random Forest*, Árvore de Decisão e KNN) e ao estado da arte em *Transformers* (BERT e RoBERTa), em três *benchmarks* consolidados: ISOT Fake News Dataset, COVID-19 Fake News Dataset e FEVER Dataset, totalizando 160.051 instâncias processadas. Os resultados demonstram que, embora os *Transformers* preservem vantagem em acurácia (até 99,79% no ISOT e 91,47% no COVID), os modelos HDC operam com complexidade linear de treinamento, mantêm desempenho competitivo no cenário balanceado (até 95,56% no ISOT) e oferecem reduções de consumo energético entre 18× e mais de 1.100× em relação ao BERT no treinamento, conforme o cenário avaliado, evidenciando *trade-off* favorável para cenários de *Big Data*, *Edge Computing* e *Green AI*. Conclui-se que a álgebra hiperdimensional não substitui os *Transformers* em precisão absoluta, mas constitui alternativa eficiente e sustentável para triagem em larga escala de desinformação, sobretudo em ambientes com restrição de recursos computacionais.

Palavras-chave: computação hiperdimensional; *fake news*; desinformação; processamento de linguagem natural; *big data*; eficiência computacional; *green AI*.

ABSTRACT

The massive dissemination of misinformation in digital environments — a phenomenon that already affected elections and public debate before 2020 and intensified during the COVID-19 pandemic — highlighted the need for automated fake news detection systems capable of operating at large scale. However, the current state of the art has been dominated by Deep Learning, Transformer, and Large Language Model (LLM) architectures, which entail high computational cost, significant energy consumption, and limited applicability in edge computing scenarios. In this context, this dissertation investigates the feasibility of Hyperdimensional Computing (HDC) as an alternative paradigm grounded in high-dimensional vector representations and lightweight algebraic operations, framed as a trade-off between predictive accuracy and energy efficiency. The research adopts a quantitative-experimental approach and evaluates the performance of five HDC variants (Vanilla HDC, AdaptHD, OnlineHD, NeuralHD, and DistHD) against classical Machine Learning baselines (Logistic Regression, SVM, Random Forest, Decision Tree, and KNN) and state-of-the-art Transformers (BERT and RoBERTa), across three consolidated benchmarks: the ISOT Fake News Dataset, the COVID-19 Fake News Dataset, and the FEVER Dataset, totaling 160,051 processed instances. Results show that, although Transformers retain an accuracy advantage (up to 99.79% on ISOT and 91.47% on COVID), HDC models exhibit linear training complexity, achieve competitive performance on the balanced scenario (up to 95.56% on ISOT), and deliver energy savings ranging from $18\times$ to over $1,100\times$ compared to BERT during training, depending on the evaluated scenario, indicating a favorable trade-off for Big Data, Edge Computing, and Green AI scenarios. It is concluded that hyperdimensional algebra does not replace Transformers in absolute precision, but constitutes an efficient and sustainable alternative for large-scale misinformation triage, particularly in resource-constrained environments.

Keywords: hyperdimensional computing; fake news; misinformation; natural language processing; big data; computational efficiency; green AI.

LISTA DE FIGURAS

Figura 1 – Taxonomia dos transtornos da informação: <i>misinformation</i> , <i>disinformation</i> e <i>malinformation</i>	23
Figura 2 – Concentração da distância de Hamming normalizada em função da dimensionalidade	26
Figura 3 – Operações básicas da Computação Hiperdimensional com hipervetores de dimensão $D = 6$	28
Figura 4 – Fluxo geral de treinamento e inferência em um classificador HDC	30
Figura 5 – Codificação de N-gramas de um texto em um único hipervetor	31
Figura 6 – Arquitetura do codificador (<i>encoder</i>) do <i>Transformer</i> , base do BERT e do RoBERTa	36
Figura 7 – Distribuição de classes nos três conjuntos de dados utilizados	68
Figura 8 – Matrizes de confusão no conjunto de dados ISOT (BoW, $D = 3000$)	75
Figura 9 – Curvas ROC no conjunto de dados ISOT (classe positiva = <i>Fake</i>)	75
Figura 10 – Matrizes de confusão no conjunto de dados COVID-19 (BoW, $D = 3000$)	75
Figura 11 – Curvas ROC no conjunto de dados COVID-19 (classe positiva = <i>Fake</i>)	76
Figura 12 – Matrizes de confusão no conjunto de dados FEVER (BoW, $D = 3000$)	76
Figura 13 – Curvas ROC no conjunto de dados FEVER (classe positiva = <i>Fake</i>)	77
Figura 14 – <i>Trade-off</i> energia $\times$ acurácia no conjunto de dados ISOT	82
Figura 15 – Acurácia e <i>F1-Score</i> por modelo no conjunto de dados ISOT	82
Figura 16 – <i>Trade-off</i> tempo de treinamento $\times$ acurácia em todos os conjuntos de dados	83
Figura 17 – Comparativo radar das variantes HDC nos três conjuntos de dados	84

LISTA DE TABELAS

Tabela 1 – Ficha Técnica e Hiperparâmetros dos Modelos Avaliados	62
Tabela 2 – Distribuição Volumétrica dos Conjuntos de Dados Utilizados	68
Tabela 3 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto ISOT	69
Tabela 4 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto COVID-19	69
Tabela 5 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto FEVER	70
Tabela 6 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto ISOT	71
Tabela 7 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto COVID-19	72
Tabela 8 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto FEVER	72
Tabela 9 – Comparação Final com <i>Transformers</i> no Conjunto ISOT (Word2Vec)	78
Tabela 10 – Comparação Final com <i>Transformers</i> no Conjunto COVID-19 (Word2Vec)	78
Tabela 11 – Comparação Final com <i>Transformers</i> no Conjunto FEVER (Word2Vec)	78
Tabela 12 – Eficiência dos Modelos Representativos de Cada Paradigma (D = 3.000, Word2Vec)	80
Tabela 13 – Custo de Inferência dos Modelos Representativos de Cada Paradigma (D = 3.000, Word2Vec)	81
Tabela A.1 – Matrizes de Confusão de Todas as Configurações (classe positiva = <i>fake</i>)	100
Tabela A.2 – Matrizes de Confusão em D = 1.000 sob Codificação TF-IDF	102

LISTA DE QUADROS

Quadro 1 – Síntese Comparativa das Variantes de HDC Avaliadas	34
Quadro 2 – Síntese dos Trabalhos Relacionados	46
Quadro 3 – Características dos Conjuntos de Dados Utilizados	50

LISTA DE ABREVIATURAS E SIGLAS

ADASYN	<i>Adaptive Synthetic Sampling</i>
AUC	<i>Area Under the Curve</i> — área sob a curva ROC
BERT	<i>Bidirectional Encoder Representations from Transformers</i>
BoW	<i>Bag of Words</i>
CNN	<i>Convolutional Neural Network</i>
COVID-19	<i>Coronavirus Disease 2019</i>
DistHD	<i>Dynamic Encoding-based Hyperdimensional Classification</i>
FEVER	<i>Fact Extraction and VERification</i> — benchmark de verificação de fatos (THORNE et al., 2018)
FN	Falso Negativo
FP	Falso Positivo
FPGA	<i>Field-Programmable Gate Array</i>
GNN	<i>Graph Neural Network</i>
GPU	<i>Graphics Processing Unit</i>
HDC	<i>Hyperdimensional Computing</i> — Computação Hiperdimensional
IA	Inteligência Artificial
IoT	<i>Internet of Things</i> — Internet das Coisas
ISOT	<i>Information Security and Object Technology Research Lab</i> (origem do conjunto de dados)
LLM	<i>Large Language Model</i> — Modelo de Linguagem de Grande Escala
LR	<i>Logistic Regression</i> — Regressão Logística
LSTM	<i>Long Short-Term Memory</i>
NeuralHD	<i>Neural Hyperdimensional Computing</i>
NLP	<i>Natural Language Processing</i> — Processamento de Linguagem Natural
OMS	Organização Mundial da Saúde
OnlineHD	<i>Online Hyperdimensional Computing</i>
PLN	Processamento de Linguagem Natural
RF	<i>Random Forest</i> — Floresta Aleatória
RoBERTa	<i>Robustly Optimized BERT Pretraining Approach</i>
ROC	<i>Receiver Operating Characteristic</i> — característica de operação do receptor
SMOTE	<i>Synthetic Minority Oversampling Technique</i>
SVM	<i>Support Vector Machine</i> — Máquina de Vetores de Suporte

TDP	<i>Thermal Design Power</i> — Potência Térmica de Projeto
TF-IDF	<i>Term Frequency–Inverse Document Frequency</i>
TN	Verdadeiro Negativo
TP	Verdadeiro Positivo
VSA	<i>Vector Symbolic Architecture</i> — Arquitetura Simbólica Vetorial
W2V	Word2Vec — Representação vetorial semântica de palavras

SUMÁRIO

1	INTRODUÇÃO	15
2	REFERENCIAL TEÓRICO	21
2.1	Desinformação Digital: Conceitos e Taxonomia	22
2.2	Computação Hiperdimensional	23
2.2.1	A Alta Dimensionalidade e as Propriedades Fundamentais	24
2.2.2	Representação de Dados	26
2.2.3	Medidas de Similaridade	27
2.2.4	Operações Básicas	27
2.2.4.1	<i>Agrupamento (Bundling)</i>	28
2.2.4.2	<i>Associação ou Ligação (Binding)</i>	28
2.2.4.3	<i>Permutação ou Rotação (Permutation)</i>	29
2.2.5	Metodologia de Aprendizagem e Classificação	29
2.2.5.1	<i>Principais Técnicas de Codificação</i>	30
2.2.5.2	<i>Codificação de N-Gramas (N-gram Encoding)</i>	30
2.2.6	Modelos Baseados em Computação Hiperdimensional	31
2.2.6.1	<i>Vanilla HDC</i>	32
2.2.6.2	<i>AdaptHD</i>	32
2.2.6.3	<i>OnlineHD</i>	32
2.2.6.4	<i>NeuralHD</i>	33
2.2.6.5	<i>DistHD</i>	33
2.2.6.6	<i>ODHD</i>	34
2.2.6.7	<i>IFODHD</i>	34
2.3	<i>Transformers</i> e o Mecanismo de Atenção	35
2.4	Representação Vetorial de Textos (<i>Embeddings</i>)	37
2.4.1	<i>Bag of Words</i> (BoW)	37
2.4.2	TF-IDF	38
2.4.3	Word2Vec	38
3	TRABALHOS RELACIONADOS	40
3.1	Aprendizado de Máquina Clássico	40
3.2	<i>Deep Learning</i> e <i>Transformers</i>	41
3.3	Verificação de Fatos e Modelagem de Evidências	43
3.4	Deteção de <i>Fake News</i> no Contexto da COVID-19	43
3.5	Computação Hiperdimensional em Tarefas de Deteção	44
3.6	Síntese e Posicionamento desta Dissertação	45
4	METODOLOGIA	48
4.1	Delineamento Metodológico	48
4.2	Coleta de Dados	49
4.2.1	Ingestão e Normalização dos Dados	51
4.2.2	Considerações Metodológicas sobre o FEVER Binarizado	51
4.2.3	Particionamento dos Conjuntos de Treinamento e Teste	53
4.3	Pré-processamento e Tratamento de Dados	54

4.4	Extração de <i>Features</i>	55
4.4.1	Representações Textuais Adotadas	55
4.4.2	Codificação Hiperdimensional (<i>Encoding</i>)	56
4.5	Modelos de Classificação	57
4.5.1	Modelos de Referência de Aprendizado de Máquina Clássico	58
4.5.2	<i>Transformers</i> : BERT e RoBERTa	58
4.5.3	Modelos de Computação Hiperdimensional	60
4.5.4	Ficha Técnica e Hiperparâmetros	61
4.6	Métricas de Avaliação	63
4.7	Estimação do Consumo Energético	64
5	EXPERIMENTOS E RESULTADOS	66
5.1	Delineamento dos Experimentos	66
5.2	Visão Geral e Consolidação dos Dados Processados	67
5.3	Experimento 1: Seleção de Dimensionalidade	68
5.4	Experimento 2: Seleção do Esquema de Codificação	71
5.5	Experimento 3: Comparação Final com <i>Transformers</i>	77
5.6	Experimento 4: Análise de Eficiência — Tempo e Energia	79
5.7	Síntese Comparativa: O <i>Trade-off</i> Acurácia × Eficiência	83
5.7.1	Complexidade de Tempo e Treinamento	83
5.7.2	Robustez ao Desbalanceamento	83
5.7.3	Mapa de Aplicabilidade	84
5.8	Limitações Observadas	85
6	CONCLUSÃO	88
6.1	Revisitação da Hipótese e Síntese dos Resultados	88
6.1.1	Validação da Eficiência Energética	89
6.1.2	Equivalência Aproximada em Cenário Balanceado (ISOT)	89
6.1.3	Limitação no Cenário Desbalanceado (COVID-19)	89
6.1.4	Limitação no Cenário de Larga Escala (FEVER)	89
6.2	Resposta aos Objetivos da Pesquisa	90
6.3	Implicações Teóricas e Práticas	91
6.4	Limitações do Estudo	91
6.5	Trabalhos Futuros	92
REFERÊNCIAS		95
APÊNDICE A – RESULTADOS COMPLETOS POR EXPERIMENTO		100

1 INTRODUÇÃO

As *fake news* podem ser definidas como conteúdo noticioso fabricado, deliberadamente falso, que imita o formato e a aparência do jornalismo legítimo com o propósito de enganar o leitor (LAZER et al., 2018). Diferentemente de erros jornalísticos involuntários, de sátiras ou de textos opinativos, caracterizam-se pela combinação de falsidade e intenção de enganar, apropriando-se da credibilidade associada ao formato jornalístico para ampliar seu alcance. Na taxonomia dos transtornos da informação de Wardle e Derakhshan (2017), detalhada na Seção 2.1, as *fake news* correspondem à forma noticiosa da *disinformation*; é nessa acepção que o termo é empregado ao longo desta dissertação.

A revolução digital e a ubiquidade das redes sociais reconfiguraram a arquitetura da comunicação global, criando um paradoxo no qual o acesso democratizado à informação coexiste com a disseminação massiva de conteúdos enganosos (VOSOUGHI; ROY; ARAL, 2018). A desinformação, contemporaneamente associada ao fenômeno das *fake news*, deixou de representar apenas um inconveniente informacional para tornar-se uma ameaça à integridade democrática, à estabilidade institucional e à saúde pública (OLAN et al., 2024). Esse cenário tornou-se ainda mais evidente durante a pandemia de COVID-19, período marcado pela intensificação da chamada “infodemia”, caracterizada pela circulação excessiva de informações falsas, imprecisas ou manipuladas em ambientes digitais (SHAHI; DIRKSON; MAJCHRZAK, 2022).

Diante desse contexto, a comunidade científica passou a propor mecanismos automatizados de detecção de *fake news* baseados em técnicas de Processamento de Linguagem Natural (PLN) e Aprendizado de Máquina (SHU et al., 2017). Inicialmente, os sistemas de classificação textual baseavam-se em análises estatísticas de conteúdo, utilizando técnicas como *N-Grams*, frequência de palavras e modelos tradicionais de aprendizado supervisionado (AHMED; TRAORE; SAAD, 2017). Posteriormente, a crescente complexidade semântica da desinformação impulsionou o desenvolvimento de arquiteturas mais sofisticadas, capazes de lidar com textos curtos, ambiguidades linguísticas e padrões contextuais mais complexos (WANG, 2017).

Nesse percurso, abordagens baseadas em *Deep Learning* passaram a incorporar arquiteturas híbridas capazes de combinar informações textuais e metadados contextuais, ampliando significativamente a capacidade de interpretação semântica dos modelos de classificação automática (RUCHANSKY; SEO; LIU, 2017; WANG, 2017). Mais

recentemente, o avanço das arquiteturas *Transformer* consolidou uma nova geração de sistemas de detecção automática de *fake news*, fundamentados em mecanismos de atenção e em representação contextual profunda da linguagem natural (VASWANI et al., 2017; DEVLIN et al., 2019).

Todavia, embora os avanços recentes tenham elevado significativamente os índices de acurácia na detecção automática de desinformação, observa-se na literatura uma relação direta entre desempenho preditivo e aumento da complexidade computacional. O atual Estado da Arte passou a ser dominado por arquiteturas massivas de *Deep Learning*, especialmente modelos baseados em *Transformers* e *Large Language Models* (LLMs), que demandam elevado poder computacional, grande capacidade de armazenamento e intenso consumo energético (STRUBELL; GANESH; MCCALLUM, 2019). Modelos de linguagem pré-treinados baseados em *Transformers* — de porte muito inferior ao dos LLMs generativos, mas ainda computacionalmente custosos —, como o BERT (*Bidirectional Encoder Representations from Transformers*) e sua evolução, o RoBERTa (*Robustly Optimized BERT Pretraining Approach*; LIU et al., 2019), apresentam desempenho superior em tarefas de classificação textual e detecção de *fake news*, sobretudo em cenários semanticamente complexos e altamente polarizados (GUNDAPU; MAMIDI, 2021), enquanto abordagens híbridas semi-supervisionadas, que integram ao texto características de usuários e conhecimento externo, ampliam ainda mais o custo associado ao treinamento e à inferência (PAKA et al., 2022).

Adicionalmente, a utilização recente de LLMs em tarefas de verificação automática de veracidade evidencia um novo paradigma baseado em modelos generativos de larga escala. Apesar do elevado potencial semântico dessas arquiteturas, sua aplicação prática enfrenta limitações relacionadas à escalabilidade, à latência, ao custo financeiro e à eficiência energética, especialmente em cenários de processamento em tempo real e computação de borda (*edge computing*) (HU et al., 2024). Define-se, assim, como problema central desta dissertação a insustentabilidade computacional dos paradigmas atuais de *Deep Learning* aplicados à triagem de desinformação em larga escala: a crescente dependência de modelos massivos e energeticamente custosos evidencia a necessidade de investigar alternativas mais leves, escaláveis e sustentáveis, capazes de oferecer um equilíbrio favorável entre acurácia preditiva e custo computacional, ainda que isso implique abrir mão de parte do desempenho absoluto dos modelos do estado da arte.

Diante desse cenário, como resposta a esse desafio, uma técnica emergente que tem ampliado as capacidades dos sistemas de classificação é a Computação Hiperdimensional (*Hyperdimensional Computing* — HDC), também conhecida como Arquitetura Vetorial Simbólica (*Vector Symbolic Architecture* — VSA), um paradigma bioinspirado que representa uma das mais promissoras convergências entre neurociência cognitiva, matemática aplicada e engenharia de sistemas (KANERVA, 2009). Esse paradigma emergiu de décadas de pesquisa interdisciplinar voltada a compreender e emular os princípios fundamentais do processamento cerebral, oferecendo uma alternativa radicalmente diferente das arquiteturas computacionais convencionais. Ao representar dados por meio de vetores de altíssima dimensionalidade — os hipervetores, que podem conter 10.000 ou mais componentes — e ao empregar operações algébricas leves e distribuídas, a HDC promete reduzir a complexidade de treinamento e inferência observada nos modelos de *Transformers* e de grafos. Cabe ressaltar que a alta dimensionalidade, por si só, não distingue o paradigma, uma vez que *embeddings* de modelos de linguagem recentes também alcançam milhares de dimensões. O que caracteriza a HDC é o uso de hipervetores gerados aleatoriamente e combinados por operações algébricas fixas, sem ajuste de pesos por retropropagação (Seção 2.2.1).

Diferentemente das técnicas tradicionais, a HDC tem sido aplicada com sucesso a tarefas como classificação, regressão e computação cognitiva, sobretudo em cenários com restrições de recursos computacionais (KLEYKO et al., 2023b). O paradigma inspira-se na hipótese de que o cérebro humano processa informações por meio de representações de alta dimensionalidade, nas quais objetos e dados são codificados como hipervetores. Nesse processo, a etapa de codificação (*encoding*) constitui o elemento central da HDC, sendo responsável por transformar os dados brutos de entrada em representações no espaço hiperdimensional, de modo a preservar simultaneamente os valores das características e suas relações de similaridade. Dessa forma, a HDC viabiliza a execução de múltiplas tarefas de aprendizado ao explorar operações simples e eficientes em espaços de alta dimensão.

Nas últimas duas décadas, o paradigma hiperdimensional consolidou-se como alternativa eficiente em diversos domínios do aprendizado de máquina, com aplicações bem-sucedidas em classificação de textos e reconhecimento de idiomas, processamento de sinais biomédicos, robótica e sistemas embarcados (KLEYKO et al., 2023a, 2023b). A formulação original do paradigma, denominada *Vanilla* HDC, constrói os protótipos de classe por simples agregação dos hipervetores de treinamento, servindo de referência para as variantes que a sucederam. No eixo da eficiência computacional, destacam-se variantes como

o AdaptHD (IMANI et al., 2019a), que propõe aprendizado adaptativo contínuo com atualização dos vetores de classe à medida que novos dados chegam; o OnlineHD (HERNÁNDEZ-CANO et al., 2021), concebido para o aprendizado incremental em passagem única, sem necessidade de múltiplas épocas de treinamento; o NeuralHD (ZOU et al., 2021), que introduz refinamento discriminativo guiado por erro de classificação, aproximando a HDC das redes neurais sem recorrer à retropropagação; e o DistHD (WANG; HUANG; IMANI, 2023), que introduz uma codificação dinâmica ciente do classificador, identificando e regenerando as dimensões que prejudicam a discriminação entre classes. Esse histórico de adoção, aliado à demonstração de treinamento eficiente a uma fração do custo das redes neurais profundas e à viabilidade do paradigma sob severa restrição de recursos, motiva a investigação de sua aplicabilidade à detecção de *fake news* em larga escala.

A hipótese investigada nesta dissertação, formulada à luz da literatura sobre *Green AI* e sobre os custos computacionais do *Deep Learning* (STRUBELL; GANESH; MCCALLUM, 2019), é a de que a Computação Hiperdimensional constitui um paradigma alternativo capaz de oferecer um *trade-off* favorável entre acurácia preditiva e eficiência energética na detecção de *fake news* em larga escala, mantendo desempenho competitivo em cenários balanceados e reduzindo o consumo computacional em ordens de magnitude em relação a arquiteturas *Transformer*. Tal formulação evita a tese mais forte — e empiricamente difícil de sustentar — de que a HDC superaria os *Transformers* em acurácia absoluta, concentrando-se no que constitui sua vantagem real: o ganho de eficiência energética para níveis aceitáveis de desempenho.

Nesse sentido, e conscientes da elevada carga computacional dos algoritmos de *Deep Learning* e dos modelos de linguagem de larga escala, propõe-se, nesta dissertação, uma investigação empírica do *trade-off* entre acurácia preditiva e eficiência computacional da Computação Hiperdimensional como paradigma algorítmico para a detecção de *fake news*, contrastando seu desempenho com o de abordagens consolidadas na literatura, de forma a identificar os cenários em que a álgebra hiperdimensional oferece vantagem prática. A investigação restringe-se à análise de dados textuais em língua inglesa e adota a classificação binária (*Fake* versus *Real*) — em que o rótulo “Real” designa o conteúdo não identificado como falso pelas fontes de anotação de cada corpus (agências de notícias no ISOT, verificadores de fatos no COVID-19 e afirmações sustentadas pela Wikipédia no FEVER), e não uma certificação absoluta de veracidade —, priorizando a triagem rápida em larga escala; a avaliação da eficiência concentra-se na complexidade algorítmica e no desempenho em

ambiente de *software*, deixando a implementação em *hardware* neuromórfico dedicado como direção para trabalhos futuros. Para tanto, adota-se uma abordagem metodológica quantitativa, experimental e comparativa, na qual variantes de classificadores hiperdimensionais são implementadas de forma controlada e contrastadas com modelos de referência consolidados e com modelos do estado da arte, sob um protocolo padronizado de avaliação, tanto em eficácia preditiva quanto em custo energético.

Para o alcance desse objetivo geral, definem-se quatro objetivos específicos: OE1 — avaliar a eficácia de classificadores hiperdimensionais na detecção de *fake news*, mensurada por acurácia, precisão, revocação, *F1-Score* e área sob a curva ROC, contrastando-os com modelos de referência de Aprendizado de Máquina clássico e com o estado da arte em *Transformers*; OE2 — verificar a robustez do paradigma em cenários de desbalanceamento de classes; OE3 — avaliar sua escalabilidade no processamento de grandes volumes de dados; e OE4 — quantificar o ganho de eficiência energética obtido em relação às arquiteturas *Transformer*, caracterizando o *trade-off* entre acurácia preditiva e custo computacional.

A pertinência desta investigação fundamenta-se na interseção entre a urgência social de mitigar a “infodemia” global e a necessidade técnica de desenvolver sistemas de Inteligência Artificial computacionalmente eficientes e passíveis de descentralização. Ao reduzir a complexidade dos métodos baseados em *Transformers* a operações lineares e paralelizáveis, a validação da HDC contribui para a agenda da *Green AI* e para a viabilidade de sistemas de Aprendizagem Federada descentralizada, direção já apontada na literatura como caminho para conciliar privacidade e eficiência (JAYAKODY; MOHAMMAD; HALGAMUGE, 2022).

As contribuições deste trabalho podem ser sintetizadas nos seguintes aportes técnicos e metodológicos:

(i) implementação de cinco variantes de classificadores hiperdimensionais (*Vanilla* HDC, *AdaptHD*, *OnlineHD*, *NeuralHD* e *DistHD*) aplicadas à tarefa de detecção de *fake news*;

(ii) integração de três representações textuais (*Bag of Words*, *TF-IDF* e *Word2Vec*) ao *pipeline* de codificação hiperdimensional, permitindo avaliar o impacto do *embedding* sobre o desempenho da HDC;

(iii) preparação de uma versão binarizada do conjunto de dados FEVER — originalmente um *benchmark* de verificação de fatos (*fact-checking*), aqui empregado na classificação apenas do texto do *claim* (Seção 4.2.2) —, com 109.809 *claims* rotulados,

viabilizando a avaliação de escalabilidade em volume uma ordem de magnitude superior à do menor corpus avaliado;

(iv) avaliação sistemática dos modelos desenvolvidos frente a cinco modelos de referência de Aprendizado de Máquina clássico (*Logistic Regression*, *SVM*, *Random Forest*, *Árvore de Decisão* e *KNN*) e a dois modelos do estado da arte (*BERT* e *RoBERTa*), em três conjuntos de dados de referência (*ISOT*, *COVID-19* e *FEVER*), totalizando 160.051 instâncias processadas;

(v) quantificação do *trade-off* entre acurácia preditiva e eficiência energética, incluindo um mapa de aplicabilidade que indica os cenários em que a HDC oferece vantagem prática sobre os *Transformers*.

O restante desta dissertação está estruturado da seguinte forma: o Capítulo 2 expõe o referencial teórico que sustenta o trabalho, abordando os conceitos e a taxonomia da desinformação digital, a Computação Hiperdimensional — suas propriedades, representações, operações, metodologia de aprendizagem e modelos derivados —, a arquitetura *Transformer* e seu mecanismo de atenção, e as técnicas de representação vetorial de textos (*embeddings*).

O Capítulo 3 revisa e analisa criticamente os trabalhos relacionados, cobrindo abordagens de Aprendizado de Máquina clássico, *Deep Learning* e *Transformers*, modelos aplicados ao contexto da COVID-19 e aplicações da Computação Hiperdimensional em tarefas de detecção, posicionando esta dissertação frente à lacuna identificada na literatura.

O Capítulo 4 descreve a metodologia adotada, organizada segundo as etapas canônicas de Aprendizado de Máquina: coleta de dados, pré-processamento, extração de *features*, modelos de classificação e seus hiperparâmetros, métricas de avaliação e estimação do consumo energético.

O Capítulo 5 apresenta os experimentos conduzidos e os resultados obtidos, interpretando o *trade-off* observado entre acurácia e eficiência e discutindo as limitações encontradas.

Por fim, o Capítulo 6 revisita a hipótese da dissertação, sintetiza os principais resultados, responde aos objetivos, discute implicações e limitações e aponta direções para pesquisas futuras. O Apêndice A consolida os resultados completos por experimento, com as quatro métricas clássicas de classificação para todas as configurações avaliadas.

2 REFERENCIAL TEÓRICO

Este capítulo apresenta o Referencial Teórico que sustenta esta pesquisa, integrando os fundamentos conceituais necessários à compreensão da proposta: os conceitos e a taxonomia da desinformação digital, a Computação Hiperdimensional (HDC), a arquitetura *Transformer* e as técnicas de representação vetorial de textos.

Inicialmente, abordam-se os conceitos e a taxonomia da desinformação digital, estabelecendo a taxonomia que organiza o fenômeno — a distinção entre *misinformation*, *disinformation* e *malinformation* — e delimitando o escopo da detecção automática de *fake news* como uma tarefa de classificação de veracidade. Discute-se, ainda, o agravamento desse cenário durante a pandemia de COVID-19, quando a chamada “infodemia” transformou a detecção em um desafio de *Big Data* de alta velocidade, no qual a verificação manual se torna inviável.

Na sequência, introduz-se a Computação Hiperdimensional, um paradigma bioinspirado no processamento distribuído do cérebro humano, que utiliza vetores de altíssima dimensionalidade para representar e manipular dados. São examinados seus princípios fundamentais — como a quase-ortogonalidade inerente a espaços de alta dimensão e a robustez a ruídos —, a representação de dados por meio de hipervetores, as medidas de similaridade empregadas e as operações algébricas básicas que sustentam o paradigma: agrupamento (*bundling*), associação (*binding*) e permutação (*permutation*). Descrevem-se, ainda, a metodologia de aprendizagem e classificação e os principais modelos derivados do paradigma (*Vanilla* HDC, *AdaptHD*, *OnlineHD*, *NeuralHD* e *DistHD*) avaliados neste trabalho.

Em seguida, caracteriza-se a arquitetura *Transformer* e seu mecanismo de atenção, fundamentos indispensáveis à compreensão dos modelos de linguagem BERT e RoBERTa, adotados como referência do estado da arte no estudo empírico comparativo.

Por fim, apresentam-se as técnicas de representação vetorial de textos (*embeddings*) — *Bag of Words*, TF-IDF e *Word2Vec* —, responsáveis por converter o conteúdo textual em vetores numéricos que alimentam tanto os classificadores hiperdimensionais quanto os modelos de referência de Aprendizado de Máquina clássico.

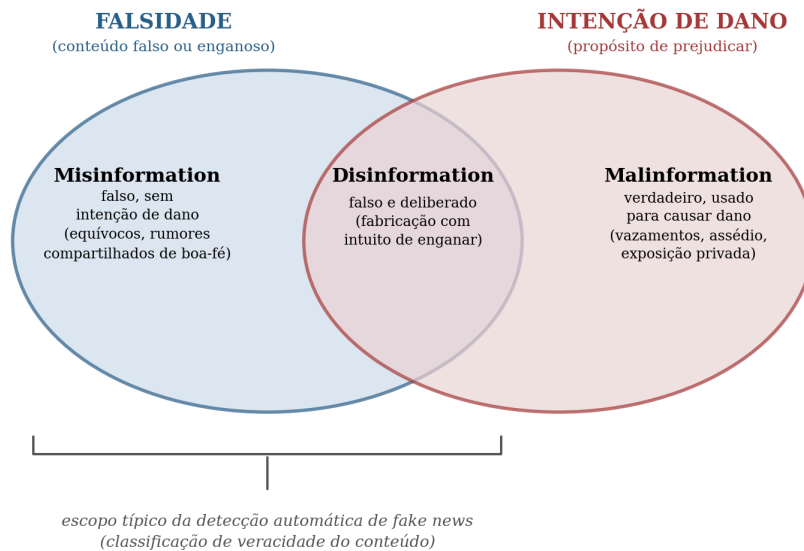
A compreensão desses conceitos é essencial para o desenvolvimento e a avaliação da proposta desta pesquisa, cujos detalhes metodológicos serão apresentados no Capítulo 4.

2.1 Desinformação Digital: Conceitos e Taxonomia

A desinformação não constitui um fenômeno único, mas um espectro de fenômenos informacionais. O referencial mais influente para organizar esse espectro é o de Wardle e Derakhshan (2017), que propõem o conceito de *transtornos da informação* (*information disorder*) e classificam os fenômenos segundo dois eixos independentes: a *falsidade do conteúdo* e a *intenção de causar dano*. Do cruzamento desses eixos emergem três categorias distintas. A *misinformation* corresponde à informação falsa disseminada sem intenção de dano — por exemplo, um usuário que compartilha de boa-fé um rumor incorreto sobre um medicamento, movido por viés de confirmação ou desconhecimento. Na prática, porém, a intenção do emissor raramente é observável: em geral, não há como determinar, a partir do conteúdo, se ele foi compartilhado de boa-fé — limitação cujas consequências para a detecção automática são discutidas a seguir. A *disinformation* corresponde à informação falsa criada e distribuída deliberadamente para enganar, manipular a opinião pública ou obter ganhos políticos e financeiros — é nesta categoria que se enquadram as *fake news* em sua acepção estrita, entendidas como conteúdo noticioso fabricado que imita o formato jornalístico. Por fim, a *malinformation* corresponde à informação verdadeira utilizada com intenção de causar dano, como vazamentos de dados privados, discurso de ódio baseado em fatos reais e campanhas de assédio (WARDLE; DERAKHSHAN, 2017).

A Figura 1 ilustra essa taxonomia e posiciona o escopo da detecção automática de *fake news*. Como os sistemas computacionais de detecção operam predominantemente sobre o conteúdo textual — e a intenção do emissor raramente é inferível a partir do texto isolado —, a tarefa é, na prática, formulada como uma classificação de veracidade: o classificador aprende a distinguir conteúdo falso de conteúdo legítimo, cobrindo tanto a *misinformation* quanto a *disinformation* — para o classificador, a distinção entre boa e má-fé não altera o rótulo atribuído, ainda que permaneça relevante para as respostas institucionais ao fenômeno, como a moderação de conteúdo e a responsabilização dos emissores. A *malinformation*, por envolver conteúdo verdadeiro, escapa ao alcance de classificadores baseados em veracidade e demanda abordagens complementares, como análise de contexto e de propagação. É nessa formulação de classificação de veracidade que se insere esta dissertação.

Figura 1 – Taxonomia dos transtornos da informação: *misinformation*, *disinformation* e *malinformation*.



Fonte: Elaborado pelo autor (2026), com base em Wardle e Derakhshan (2017).

A complexidade deste ecossistema atingiu um ponto de inflexão durante a pandemia de COVID-19. O termo “infodemia”, cunhado pela Organização Mundial da Saúde, descreve a saturação do espaço informacional por uma mistura de fatos, rumores, pseudociência e teorias da conspiração. Como observam Shahi, Dirkson e Majchrzak (2022), a infodemia transformou a detecção de *fake news* em um desafio de *Big Data* de alta velocidade, em que a verificação manual se torna inviável. A detecção automática neste cenário exige que os algoritmos compreendam não apenas a sintaxe, mas o contexto semântico de termos médicos e a dinâmica de polarização social em plataformas como o Twitter.

2.2 Computação Hiperdimensional

A premissa central do HDC é uma observação da neurociência: o cérebro não computa com números escalares ou símbolos isolados, mas com padrões de atividade distribuídos por circuitos massivos, compostos por bilhões de neurônios e sinapses (KANERVA, 2009). Inspirado nessa arquitetura, o HDC representa qualquer informação — uma letra, uma palavra, um documento inteiro — por meio de *hipervetores*: vetores pseudoaleatórios de dimensionalidade D muito elevada, tipicamente na ordem de milhares de posições (por exemplo, $D = 3.000$). Nesses vetores, a informação é distribuída e redundante, de modo análogo ao cérebro, que tolera a falha de neurônios individuais sem perder a memória como um todo.

Uma intuição útil para o leitor não familiarizado é pensar no HDC como uma “aritmética de conceitos”. Cada conceito recebe um hipervetor, e um conjunto reduzido de operações algébricas — descritas na Seção 2.2.4 — permite combinar, associar e recuperar conceitos, formando uma memória associativa: um banco de dados de padrões em que a busca é feita por semelhança, e não por endereço exato. Enquanto a computação clássica manipula bits individuais em campos estruturados, e as redes neurais profundas ajustam milhões de pesos por retropropagação, o HDC aprende construindo protótipos de classe por meio de somas de hipervetores, o que reduz drasticamente a complexidade de treinamento e favorece a paralelização (KLEYKO et al., 2023a).

No contexto do aprendizado de máquina, o objetivo do paradigma é oferecer classificadores leves, tolerantes a ruído e de complexidade linear, adequados a cenários com grandes volumes de dados ou com restrições de energia e processamento — exatamente o regime de interesse desta dissertação.

2.2.1 A Alta Dimensionalidade e as Propriedades Fundamentais

O termo *hiperespaço* denota o espaço de dimensionalidade muito elevada em que os hipervetores habitam. Kanerva (2009) destaca três propriedades desse espaço que sustentam todo o paradigma: a robustez, a representação distribuída e a aleatoriedade.

A **robustez** decorre da redundância: mesmo que parte das posições de um hipervetor seja corrompida, perdida ou invertida, a informação original ainda pode ser reconhecida — de forma análoga aos códigos corretores de erro das comunicações digitais, que toleram certo número de bits errados. Quanto maior a dimensão do espaço, maior a quantidade de posições em que dois padrões podem diferir sem se confundirem, ou seja, a tolerância a erros cresce com a dimensionalidade.

A **representação distribuída** (ou holográfica) significa que a informação não reside em uma posição específica do vetor, mas está espalhada homogeneamente por todas as suas dimensões. Se alguns bits falham, a informação armazenada se degrada de forma gradual e proporcional, independentemente da localização das falhas — em contraste com a codificação convencional, na qual os bits são organizados em campos posicionais e a corrupção de um único campo pode destruir o dado.

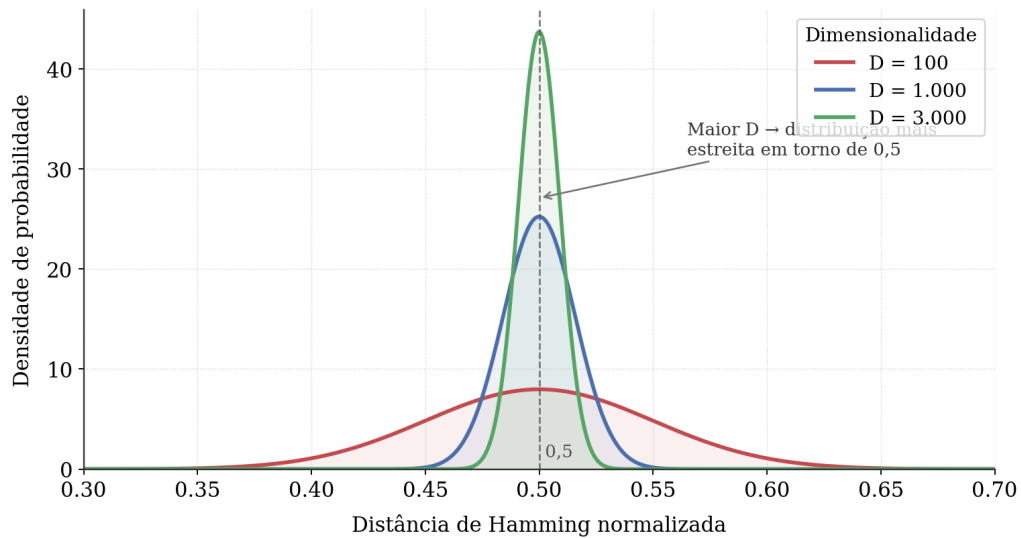
A **aleatoriedade**, por fim, é o mecanismo que garante a distinguibilidade dos conceitos. Cada novo símbolo recebe um hipervetor gerado aleatoriamente (por exemplo, cada bit sorteado com probabilidade de 50%), e uma propriedade estatística notável dos

espaços de alta dimensão — a **forte tendência de vetores aleatórios à quase-ortogonalidade** — assegura que dois hipervetores gerados aleatoriamente sejam, com probabilidade próxima de 1, aproximadamente ortogonais entre si, isto é, não relacionados. Formalmente, para vetores binários de dimensão D , a distância de Hamming esperada entre dois hipervetores aleatórios é:

$$E[d_Hamming(A, B)] = D / 2, \quad Var \approx D / 4$$

Em termos concretos: para $D = 3.000$, dois hipervetores aleatórios diferem, em média, em 1.500 posições, e a distância de Hamming normalizada (dividida por D) concentra-se fortemente em torno de 0,5. A Figura 2 ilustra esse fenômeno: à medida que a dimensionalidade cresce de 100 para 3.000, a distribuição da distância normalizada torna-se progressivamente mais estreita ao redor da média. Isso significa que o sistema pode alocar milhares de conceitos distintos no mesmo espaço vetorial sem que eles interfiram significativamente uns nos outros (*crosstalk* mínimo), diferentemente de *embeddings* densos de baixa dimensão (como *Word2Vec*, de 300 dimensões), que exigem otimização cuidadosa para evitar sobreposição semântica. A diferença, contudo, não é apenas de dimensão: os *embeddings* densos são aprendidos por otimização justamente para que palavras relacionadas ocupem posições próximas — isto é, são deliberadamente correlacionados —, ao passo que os hipervetores atômicos do HDC são aleatórios e não carregam semântica prévia. É essa independência inicial, garantida pela quase-ortogonalidade, que permite ao HDC compor estruturas por *bundling* e *binding* sem interferência; mesmo *embeddings* de alta dimensão, como os de modelos de linguagem recentes, não gozam dessa propriedade.

Figura 2 – Concentração da distância de Hamming normalizada em função da dimensionalidade.



Fonte: Elaborado pelo autor (2026).

2.2.2 Representação de Dados

O princípio que rege a codificação no HDC é simples de enunciar: entradas semelhantes (ou da mesma classe) devem ser mapeadas para hipervetores próximos no hiperespaço, e entradas não relacionadas devem ser mapeadas para hipervetores aproximadamente ortogonais. Esse princípio guarda paralelo direto com a hipótese distribucional que fundamenta os *embeddings* densos (Seção 2.4.3), segundo a qual itens que ocorrem em contextos semelhantes devem ocupar regiões próximas do espaço vetorial. A diferença está no mecanismo: nos *embeddings*, a proximidade é aprendida por otimização; no HDC, ela é construída pela composição algébrica de hipervetores aleatórios. A literatura consolidou diferentes famílias de representação, que variam conforme o domínio numérico dos componentes do vetor (KLEYKO et al., 2023a): hipervetores binários, com componentes em $\{0, 1\}$; hipervetores bipolares, com componentes em $\{-1, +1\}$; hipervetores reais, como nas Representações Holográficas Reduzidas (HRR) de Plate (2003); e hipervetores complexos, na extensão de Fourier das HRR.

A escolha da representação envolve um compromisso prático: hipervetores binários e bipolares admitem operações extremamente baratas (portas lógicas e somas inteiras), sendo preferidos em implementações de *hardware* e em cenários com restrição de recursos; hipervetores reais preservam informação mais fina e tendem a oferecer maior precisão, ao custo de aritmética de ponto flutuante (KLEYKO et al., 2023a, 2023b). Em todos os casos, as

propriedades da Seção 2.2.1 se mantêm: a alta dimensionalidade garante a quase ortogonalidade de vetores aleatórios, independentemente do domínio numérico adotado.

2.2.3 Medidas de Similaridade

Toda decisão em um sistema HDC — inclusive a classificação final — reduz-se a uma pergunta: quão parecidos são dois hipervetores? Para respondê-la, empregam-se métricas de similaridade adequadas ao tipo de representação. Para hipervetores binários, utiliza-se a **distância de Hamming**, que conta o número de posições em que dois vetores diferem (calculada eficientemente pela operação XOR bit a bit). Em sua forma normalizada (dividida por D), assume valor 0 para vetores idênticos, aproximadamente 0,5 para vetores não relacionados (ortogonais) e valores próximos de 1 para vetores complementares.

Um exemplo minúsculo torna a ideia concreta. Considerem-se dois vetores de dimensão $D = 6$: $A = [1, 0, 1, 0, 1, 1]$ e $B = [0, 1, 1, 0, 0, 1]$. Eles diferem nas posições 1, 2 e 5, logo $d_{\text{Hamming}}(A, B) = 3$ e a distância normalizada é $3/6 = 0,5$ — exatamente o valor esperado para vetores não relacionados. Em dimensões realistas ($D = 3.000$), esse mesmo cálculo distingue com enorme confiança pares relacionados (distância bem abaixo de 0,5) de pares aleatórios (distância concentrada em 0,5), conforme a Figura 2.

Para hipervetores reais ou bipolares, a métrica usual é a **similaridade do cosseno**:

$$\cos(\theta) = \langle a, b \rangle / (\|a\| \cdot \|b\|)$$

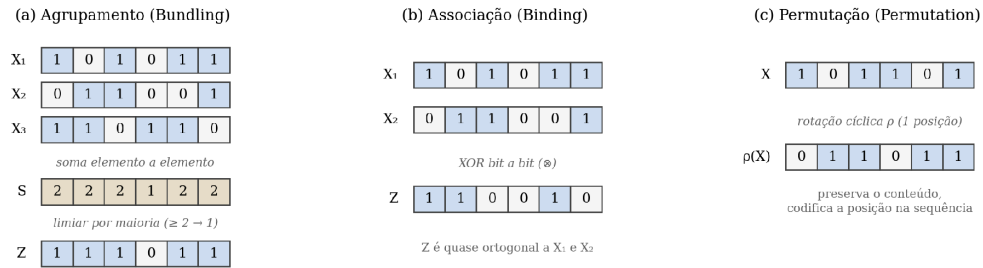
que mede o cosseno do ângulo entre os vetores e, por ser independente da magnitude, é apropriada quando os componentes assumem valores contínuos. O produto interno simples $\langle a, b \rangle$ também é empregado quando os vetores são normalizados previamente. Em um classificador HDC, a amostra é atribuída à classe cujo hipervetor de referência apresenta a maior similaridade (ou a menor distância) com o hipervetor da amostra.

2.2.4 Operações Básicas

O processamento de informação no HDC ocorre por meio de três operações algébricas elementares, frequentemente resumidas pelo acrônimo MAP (*Multiply–Add–Permute*): a associação (*binding*), o agrupamento (*bundling*) e a permutação (*permutation*). Todas preservam a dimensionalidade — o resultado é sempre um hipervetor do mesmo espaço dos operandos — e podem ser compostas livremente, o que permite codificar estruturas de dados complexas (pares atributo-valor, sequências, textos inteiros) em um único hipervetor holístico.

A Figura 3 ilustra as três operações com vetores binários de dimensão $D = 6$, propositalmente pequenos para fins didáticos; as subseções a seguir detalham cada uma delas.

Figura 3 – Operações básicas da Computação Hiperdimensional com hipervetores de dimensão $D = 6$.



Fonte: Elaborado pelo autor (2026).

2.2.4.1 Agrupamento (Bundling)

O agrupamento ($\oplus$) combina um conjunto de hipervetores $\{X_1, X_2, \dots, X_n\}$ em um único hipervetor Z que é simultaneamente similar a todos os operandos — funcionando como uma “memória coletiva” do conjunto. Em vetores binários, implementa-se pela soma elemento a elemento seguida de uma função de limiar por maioria: cada posição do vetor resultante recebe 1 se a maioria dos operandos tiver 1 naquela posição, e 0 caso contrário.

No exemplo da Figura 3(a), com $X_1 = [1, 0, 1, 0, 1, 1]$, $X_2 = [0, 1, 1, 0, 0, 1]$ e $X_3 = [1, 1, 0, 1, 1, 0]$, a soma elemento a elemento produz $S = [2, 2, 2, 1, 2, 2]$; como há três operandos, a maioria exige pelo menos dois “votos”, e a limiarização resulta em $Z = [1, 1, 1, 0, 1, 1]$. O vetor Z apresenta distância de Hamming pequena em relação a cada operando — muito menor do que a esperada para vetores aleatórios —, o que permite reconhecer, mais tarde, que ele representa aquela combinação específica. É por meio do *bundling* que se constrói o protótipo de uma classe (por exemplo, “fake”): somam-se os hipervetores de todas as amostras de treinamento daquela classe.

2.2.4.2 Associação ou Ligação (Binding)

A associação ($\otimes$) vincula dois hipervetores que guardam relação semântica entre si — por exemplo, uma variável e seu valor, ou uma posição e um símbolo. Em vetores binários, é implementada pela operação XOR (ou-exclusivo) elemento a elemento:

$$A \otimes B = A \text{ XOR } B, \text{ com } (A \otimes B) \otimes B = A \text{ (auto-inversa)}$$

Duas propriedades tornam o *binding* especialmente útil. Primeiro, o vetor resultante é aproximadamente ortogonal a ambos os operandos — no exemplo da Figura 3(b), $Z = X_1 \otimes X_2$

$= [1, 1, 0, 0, 1, 0]$ dista de X_1 e de X_2 o mesmo que um vetor aleatório —, o que preserva a distinguibilidade entre o par ligado e seus componentes. Segundo, a operação é auto-inversa: aplicar novamente o XOR com um dos operandos recupera o outro, permitindo “desligar” (*unbinding*) e consultar a informação armazenada.

2.2.4.3 Permutação ou Rotação (*Permutation*)

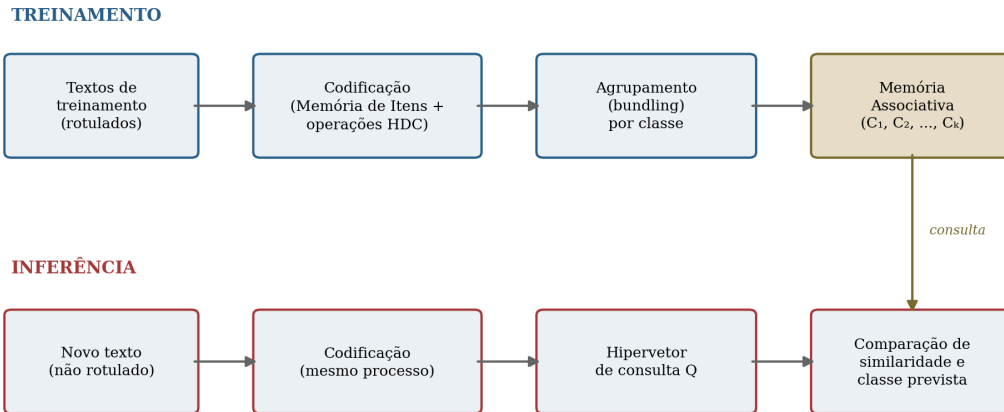
A permutação (ρ) embaralha as posições de um hipervetor segundo um padrão fixo — tipicamente uma rotação cíclica dos bits —, produzindo um vetor quase ortogonal ao original. Sua função é codificar ordem e sequencialidade: aplicando ρ zero, uma, duas ou mais vezes, um mesmo símbolo passa a ser representado por vetores distintos conforme a posição que ocupa na sequência. No exemplo da Figura 3(c), a rotação de $X = [1, 0, 1, 1, 0, 1]$ em uma posição produz $\rho(X) = [0, 1, 1, 0, 1, 1]$. É essa operação que permite distinguir, por exemplo, a sequência “A causa B” de “B causa A” — informação essencial para a análise de N-gramas descrita na Seção 2.2.5.2.

2.2.5 Metodologia de Aprendizagem e Classificação

Combinadas, as operações anteriores dão origem a um fluxo completo de aprendizado supervisionado, ilustrado na Figura 4 e composto por três fases. Na fase de **codificação**, cada amostra de entrada é transformada em um hipervetor por meio de hipervetores atômicos pré-armazenados na chamada Memória de Itens e das operações de *binding*, *bundling* e permutação. Na fase de **treinamento**, os hipervetores das amostras de cada classe são agrupados (*bundling*), produzindo k hipervetores de classe $\{C_1, C_2, \dots, C_k\}$ — os protótipos — armazenados na Memória Associativa. Na fase de **inferência**, uma nova amostra é codificada pelo mesmo processo, gerando um hipervetor de consulta Q , que é comparado com todos os protótipos por meio de uma medida de similaridade; o rótulo do protótipo mais similar é retornado como previsão.

Vale destacar o que esse fluxo não contém: não há retropropagação, não há ajuste iterativo de milhões de pesos, e o treinamento básico pode ser realizado em uma única passagem sobre os dados. Cada amostra contribui para o protótipo de sua classe com um custo constante, o que confere ao treinamento complexidade linear no número de amostras — a origem da eficiência computacional que motiva esta dissertação.

Figura 4 – Fluxo geral de treinamento e inferência em um classificador HDC.



Fonte: Elaborado pelo autor (2026).

2.2.5.1 Principais Técnicas de Codificação

De todas as etapas, a codificação é provavelmente a mais determinante para a qualidade do classificador (KLEYKO et al., 2023b). O ponto de partida são os hipervetores base da Memória de Itens, que podem ser de três tipos. Os **hipervetores aleatórios** são gerados aleatoriamente e, pela propriedade da quase ortogonalidade, prestam-se a representar dados categóricos e símbolos discretos (letras, palavras, rótulos). Os **hipervetores de nível** preservam correlação entre valores vizinhos e são usados para grandezas ordenadas (idade, peso, frequências): constrói-se um espectro $\{L_1, \dots, L_m\}$ em que níveis adjacentes compartilham alta similaridade e níveis extremos são quase ortogonais. Os **hipervetores circulares**, por fim, capturam padrões cíclicos, como horas do dia ou estações do ano.

Sobre esses vetores base, a literatura consolida duas estratégias canônicas de composição. Na codificação baseada em registros (*record-based encoding*), cada atributo i é representado pelo *binding* entre um hipervetor de posição ID_i e o hipervetor de nível correspondente ao seu valor, e os pares resultantes são agregados por *bundling*: $H = \sum (ID_i \otimes L_i)$. Essa estratégia é natural para dados tabulares. Para dados sequenciais — como texto —, a estratégia mais adequada é a codificação de N-gramas, detalhada a seguir.

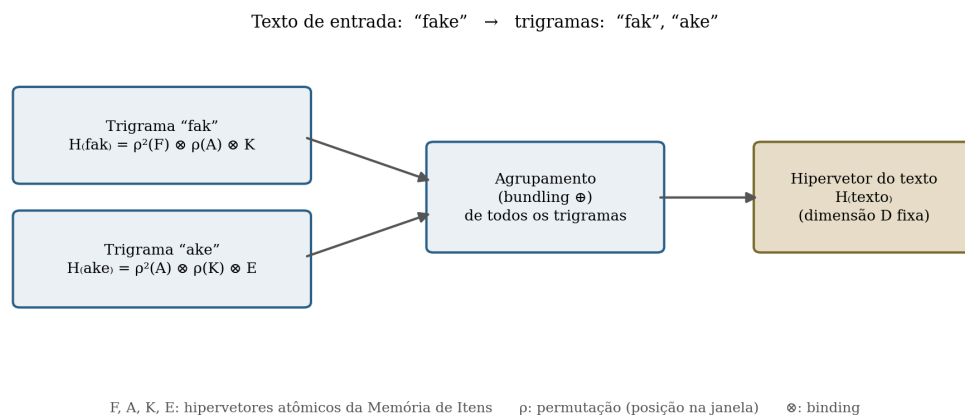
2.2.5.2 Codificação de N-Gramas (N-gram Encoding)

Proposta por Joshi, Halseth e Kanerva (2017) para modelar linguagem, a codificação de N-gramas representa a ordem sequencial dos símbolos por meio de permutações, dispensando vetores de identidade explícitos. O processo, ilustrado na Figura 5, opera em três etapas: (i) cada símbolo do alfabeto (por exemplo, cada caractere) recebe um hipervetor atômico aleatório na Memória de Itens; (ii) para cada janela de n símbolos consecutivos (um

n-grama), o hipervetor do símbolo na posição j sofre $(n - j)$ permutações rotacionais, e os vetores permutados são combinados por *binding*, produzindo um hipervetor único para aquele n-grama; (iii) os hipervetores de todos os n-gramas do texto são agregados por *bundling*, resultando em um único hipervetor H de dimensão fixa D que representa o texto inteiro.

Um exemplo concreto: para a palavra “fake” com trigramas ($n = 3$), extraem-se as janelas “fak” e “ake”. O trigramma “fak” é codificado como $H(\text{fak}) = \rho^2(F) \otimes \rho(A) \otimes K$, em que F , A , K e E denotam os hipervetores atômicos associados, respectivamente, aos caracteres “f”, “a”, “k” e “e”, e ρ denota a rotação cíclica; analogamente, $H(\text{ake}) = \rho^2(A) \otimes \rho(K) \otimes E$. O hipervetor do texto é $H(\text{texto}) = H(\text{fak}) \oplus H(\text{ake})$. Note-se que a permutação garante que a letra “a” na segunda posição de “fak” e na primeira posição de “ake” produza contribuições distintas, preservando a informação de ordem. Duas propriedades tornam essa codificação atraente para a detecção de *fake news*: a dimensão do vetor resultante é fixa, independentemente do comprimento do texto, e os padrões estatísticos locais capturados pelos N-gramas dispensam qualquer etapa de treinamento do codificador. Cabe observar que o exemplo acima emprega n-gramas de caracteres por clareza didática, mas a mesma mecânica se aplica a n-gramas de palavras, bastando que cada termo do vocabulário — e não cada letra — receba um hipervetor atômico na Memória de Itens. Essa variante não é, contudo, a adotada nesta dissertação, cujos codificadores são descritos na Seção 4.4.2; a codificação de N-gramas é aqui apresentada por constituir a estratégia canônica do paradigma para dados sequenciais.

Figura 5 – Codificação de N-gramas de um texto em um único hipervetor.



Fonte: Elaborado pelo autor (2026).

2.2.6 Modelos Baseados em Computação Hiperdimensional

Sobre o arcabouço descrito nas seções anteriores, a literatura desenvolveu variantes que se diferenciam, sobretudo, quanto à estratégia de aprendizado — isto é, quanto à forma de construir e refinar os hipervetores de classe. Esta seção apresenta as variantes mais proeminentes, em ordem crescente de sofisticação do mecanismo de atualização, e encerra-se com dois modelos que aplicam o paradigma à detecção de anomalias, linha de pesquisa diretamente relacionada ao problema investigado nesta dissertação.

2.2.6.1 *Vanilla HDC*

O *Vanilla* HDC é a arquitetura clássica do paradigma. Seu treinamento consiste na soma direta (*bundling*) dos hipervetores pertencentes à mesma classe, formando protótipos representativos sem qualquer mecanismo de correção de erro: cada amostra contribui igualmente para o protótipo, independentemente de o modelo já classificá-la corretamente ou não. Essa simplicidade lhe confere extrema eficiência computacional e treinamento em passagem única, mas o expõe à saturação dos protótipos em cenários semanticamente complexos: amostras majoritárias ou redundantes dominam a soma, e a fronteira de decisão perde poder discriminativo (KANERVA, 2009; KLEYKO et al., 2023b).

2.2.6.2 *AdaptHD*

O *AdaptHD*, proposto por Imani et al. (2019a), introduz aprendizado iterativo com correção de erro inspirado no Perceptron: sempre que uma amostra é classificada incorretamente, seu hipervetor é adicionado ao protótipo da classe correta e subtraído do protótipo da classe incorreta, aumentando progressivamente a separabilidade entre as categorias no hiperespaço. O modelo adota, ainda, taxas de aprendizado adaptativas com dois mecanismos complementares: um dependente da iteração, que aplica taxa elevada nas primeiras épocas para acelerar a convergência e a reduz gradualmente, e um dependente dos dados, que intensifica a atualização nas amostras com maior erro. O resultado é uma convergência substancialmente mais rápida do que a do treinamento iterativo ingênuo.

2.2.6.3 *OnlineHD*

O *OnlineHD*, introduzido por Hernández-Cano et al. (2021), é voltado ao aprendizado contínuo em fluxo de dados e opera em passagem única. Sua inovação central é ponderar a atualização de cada amostra pela similaridade com os protótipos existentes: amostras muito semelhantes ao que o modelo já aprendeu produzem atualizações mínimas, enquanto amostras novas ou surpreendentes recebem peso elevado. Essa estratégia evita a saturação dos protótipos — o principal defeito do *Vanilla* HDC — e torna o modelo especialmente eficiente

em cenários desbalanceados e em aplicações de *streaming*, mantendo acurácia comparável à de treinamentos com múltiplas épocas.

2.2.6.4 *NeuralHD*

O NeuralHD (ZOU et al., 2021) inspira-se na plasticidade e na regeneração neuronal do cérebro humano. Durante o treinamento, o modelo monitora o poder discriminativo de cada dimensão dos hipervetores e regenera seletivamente as dimensões pouco informativas, substituindo-as por novas projeções. Combinada a uma codificação não linear, essa regeneração dinâmica aumenta a capacidade de separação semântica sem elevar a dimensionalidade total, além de conferir robustez adicional a ruídos — características que tornam o modelo adequado a dispositivos de borda com restrições de memória e energia.

2.2.6.5 *DistHD*

O DistHD (WANG; HUANG; IMANI, 2023) combina dois mecanismos: uma codificação dinâmica que identifica e regenera as dimensões que contribuem negativamente para a acurácia, e um critério de decisão top-2, no qual a classificação considera as duas classes mais similares à amostra, aumentando a robustez em regiões de sobreposição entre classes. O modelo foi desenhado para alta paralelização, acelerando treinamento e inferência e reduzindo gargalos de memória — o que o torna particularmente atraente para cenários de *Big Data* e computação distribuída.

Variantes adicionais — como BinHD (KIM; IMANI; ROSING, 2018), CompHD (IMANI et al., 2019c), SparseHD (SALAMAT; IMANI; ROSING, 2019), QuantHD (IMANI et al., 2019b) e LeHDC (DUAN et al., 2022) — exploram, respectivamente, binarização completa, compressão de modelos, esparsidade, quantização e aprendizado baseado em gradiente, e são revisadas em profundidade nos levantamentos de Kleyko et al. (2023a, 2023b).

O Quadro 1 sintetiza como as cinco variantes avaliadas nesta dissertação se comparam quanto ao mecanismo de aprendizado, ao número de passagens sobre os dados, à principal vantagem e ao cenário de aplicação para o qual cada uma foi proposta.

Quadro 1 – Síntese Comparativa das Variantes de HDC Avaliadas

Variante	Mecanismo de aprendizado	Passagens na proposta original	Principal vantagem	Cenário de aplicação proposto
<i>Vanilla</i> HDC (KANERVA, 2009)	Soma direta (<i>bundling</i>) dos hipervetores de cada classe, sem correção de erro	Única	Máxima simplicidade e custo mínimo	Referência (<i>baseline</i>) do paradigma; protótipos saturam em dados complexos
AdaptHD (IMANI et al., 2019a)	Correção iterativa de erro inspirada no Perceptron, com taxa de aprendizado adaptativa	Múltiplas	Convergência mais rápida que o treinamento iterativo ingênuo	Tarefas em que a acurácia do <i>Vanilla</i> HDC é insuficiente
OnlineHD (HERNÁNDEZ-CANO et al., 2021)	Atualização ponderada pela similaridade com os protótipos existentes	Única	Evita a saturação dos protótipos	Aprendizado contínuo em fluxo de dados (<i>streaming</i>) e dados desbalanceados
NeuralHD (ZOU et al., 2021)	Regeneração das dimensões pouco discriminativas, com codificação não linear	Múltiplas	Maior separabilidade sem elevar D; robustez a ruído	Dispositivos de borda com restrição de memória e energia
DistHD (WANG; HUANG; IMANI, 2023)	Codificação dinâmica que regenera dimensões prejudiciais, com decisão <i>top-2</i>	Múltiplas	Robustez em regiões de sobreposição entre classes; alta paralelização	<i>Big Data</i> e computação distribuída

Fonte: Elaborado pelo autor (2026), com base nas referências indicadas. Nesta dissertação, todas as variantes foram executadas com 2 épocas (Seção 4.5.3).

2.2.6.6 ODHD

O ODHD (WANG; JIAO; HU, 2022) aplica a álgebra hiperdimensional à detecção de anomalias em uma formulação de aprendizado de classe única (*one-class learning*): em vez de construir um protótipo por classe, o modelo aprende exclusivamente a partir de amostras normais, agregando seus hipervetores por *bundling* em um único protótipo que representa o comportamento esperado do sistema. Na inferência, cada nova amostra é comparada a esse protótipo por medida de similaridade, e valores baixos indicam desvio do padrão aprendido. Seu principal diferencial é o mecanismo automático de seleção de limiar, que deriva a fronteira entre normalidade e anomalia da própria distribuição de similaridades do conjunto de treinamento, dispensando o ajuste manual de hiperparâmetros sensíveis. O modelo preserva a eficiência computacional e a paralelização características do paradigma, viabilizando treinamento e inferência rápidos em dispositivos de baixo consumo energético.

2.2.6.7 IFODHD

O IFODHD (XU et al., 2025) estende essa linha ao incorporar seleção de características ao processo de codificação. A proposta parte de uma limitação das formulações convencionais de HDC, nas quais todos os atributos são codificados com o mesmo peso, o

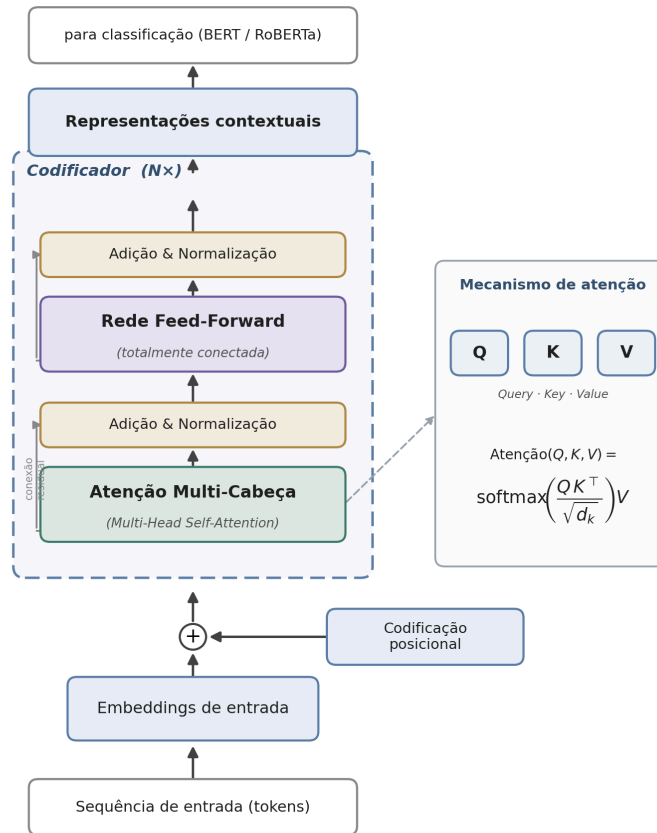
que pode introduzir ruído e reduzir a capacidade discriminativa do protótipo. O IFODHD avalia a relevância de cada atributo para distinguir padrões normais de anômalos e pondera ou seleciona apenas os mais informativos antes da projeção para o hiperespaço; sobre essa representação depurada é então construído o protótipo de normalidade usado na detecção por similaridade. Os autores reportam ganhos de acurácia e redução de falsos positivos frente a métodos HDC convencionais de detecção de *outliers*, preservando o baixo custo computacional do paradigma.

Cabe observar que o ODHD e o IFODHD não são avaliados experimentalmente nesta dissertação, cujo escopo é a classificação supervisionada de veracidade textual. São aqui apresentados por constituírem a linha de pesquisa em HDC mais próxima do problema investigado — a identificação de instâncias que desviam do padrão esperado, frequentemente sob forte desbalanceamento —, servindo de referência conceitual para a discussão dos trabalhos relacionados no Capítulo 3.

2.3 Transformers e o Mecanismo de Atenção

A arquitetura *Transformer*, proposta originalmente por Vaswani et al. (2017), substituiu completamente os mecanismos recorrentes por um sistema baseado em atenção, permitindo o processamento paralelo de sequências textuais e a captura de dependências semânticas de longo alcance. Esta arquitetura constitui a base dos modelos comparativos BERT e RoBERTa utilizados nesta dissertação. A Figura 6 sintetiza os componentes do codificador (*encoder*) — a parte da arquitetura original efetivamente empregada pelo BERT e pelo RoBERTa —, das entradas ao mecanismo de atenção; o decodificador (*decoder*), voltado à geração de texto, não é representado por não integrar os modelos avaliados.

Figura 6 – Arquitetura do codificador (*encoder*) do *Transformer*, base do BERT e do RoBERTa.



Fonte: Elaborado pelo autor (2026), adaptado de Vaswani et al. (2017).

O principal mecanismo responsável pelo desempenho dos *Transformers* é o **Self-Attention**. Esta técnica atribui pesos dinâmicos aos *tokens* de uma sequência conforme sua relevância contextual, calculando relações internas entre todos os *tokens* simultaneamente. Formalmente, o mecanismo utiliza três projeções vetoriais — *Query* (Q), *Key* (K) e *Value* (V) — sendo a atenção calculada como:

$$Attention(Q, K, V) = softmax(Q \cdot K^T / \sqrt{d_k}) \cdot V$$

onde d_k é a dimensão dos vetores *Key* e o fator de normalização $\sqrt{d_k}$ evita instabilidades numéricas durante o treinamento. Em tarefas de detecção de *fake news*, esta capacidade é particularmente importante porque conteúdos falsos frequentemente utilizam ambiguidades semânticas, manipulação discursiva e estruturas argumentativas complexas.

A partir desta arquitetura surgiram modelos pré-treinados de larga escala, entre os quais o **BERT** (*Bidirectional Encoder Representations from Transformers*), desenvolvido por Devlin et al. (2019). O BERT introduziu o conceito de treinamento bidirecional profundo, permitindo que o modelo compreendesse simultaneamente o contexto à esquerda e à direita de

cada palavra. Seu treinamento ocorre principalmente através das tarefas de *Masked Language Modeling* (MLM) e *Next Sentence Prediction* (NSP).

Posteriormente, Liu et al. (2019) propuseram o **RoBERTa** (*Robustly Optimized BERT Pretraining Approach*), versão otimizada do BERT. O RoBERTa removeu a tarefa de NSP, aumentou o volume de dados de pré-treinamento e ampliou o tempo de aprendizado, alcançando desempenho superior em diversas tarefas de PLN.

Apesar da elevada acurácia, os modelos *Transformer* apresentam limitações importantes relacionadas ao elevado custo computacional. O *Self-Attention* possui complexidade quadrática em relação ao comprimento da sequência, $O(N \cdot L^2)$, em contraste com a complexidade linear $O(N \cdot L)$ das operações fundamentais do HDC. Strubell, Ganesh e McCallum (2019) estimam que, no caso extremo do desenvolvimento de um *Transformer* com busca de arquitetura neural (*neural architecture search*), o processo pode emitir CO_2 equivalente às emissões de cinco automóveis ao longo de toda a sua vida útil, motivando a busca por paradigmas alternativos.

2.4 Representação Vetorial de Textos (*Embeddings*)

Nenhum dos paradigmas apresentados neste capítulo opera diretamente sobre texto bruto: algoritmos de aprendizado de máquina manipulam números, de modo que toda tarefa de classificação textual exige uma etapa prévia de conversão do texto em uma representação numérica — processo genericamente denominado *embedding* ou vetorização. A qualidade dessa representação condiciona diretamente o desempenho do classificador, pois é ela que determina quais propriedades do texto (frequência de termos, raridade, semântica) ficam acessíveis ao modelo. As técnicas evoluíram ao longo de um espectro: das representações esparsas baseadas em frequência (*Bag of Words*, TF-IDF) às representações densas com semântica distribucional (*Word2Vec*) e, mais recentemente, às representações contextuais produzidas pelos próprios *Transformers* (Seção 2.3). Esta seção detalha as três técnicas clássicas, da mais simples à mais sofisticada.

2.4.1 *Bag of Words* (BoW)

O *Bag of Words* é a representação mais elementar: o texto é tratado como um “saco de palavras” sem ordem (HARRIS, 1954). Constrói-se um vocabulário com os termos do corpus e cada documento é representado por um vetor cujo tamanho é o do vocabulário, em que cada posição registra quantas vezes o termo correspondente ocorre no documento. Um exemplo minúsculo: dado o vocabulário {vacina, cura, governo, estudo}, a frase “estudo confirma que

vacina não é cura” seria representada pelo vetor $[1, 1, 0, 1]$ — uma ocorrência de “vacina”, uma de “cura”, nenhuma de “governo” e uma de “estudo”.

A simplicidade do BoW é sua força e sua fraqueza. Por um lado, é barato de computar e surpreendentemente eficaz quando o vocabulário empregado já distingue as classes — caso frequente em *fake news*, cujo léxico sensacionalista difere do jornalístico. Por outro, a representação descarta completamente a ordem das palavras (“governo esconde cura” e “cura esconde governo” produzem o mesmo vetor), gera vetores esparsos de alta dimensionalidade (um por termo do vocabulário) e trata todos os termos com o mesmo peso, ainda que palavras muito comuns carreguem pouca informação discriminativa.

2.4.2 TF-IDF

O TF-IDF (*Term Frequency–Inverse Document Frequency*) refina o BoW ao ponderar cada termo por sua importância relativa: a frequência do termo no documento (TF) é multiplicada pelo inverso de sua frequência no corpus (IDF), de modo que termos raros — e, portanto, mais informativos — recebem peso maior, enquanto termos onipresentes são atenuados. Formalmente, para um termo t em um documento d de um corpus com N documentos:

$$tf-idf(t, d) = tf(t, d) \times \log(N / df(t))$$

onde $tf(t, d)$ é a contagem de t em d e $df(t)$ é o número de documentos que contêm t . A intuição é direta: a palavra “hydroxychloroquine”, presente em poucos documentos, é mais reveladora do conteúdo de um texto do que a palavra “day”, presente em quase todos. O TF-IDF preserva a esparsidade e a perda de ordem do BoW, mas melhora substancialmente a separabilidade das classes em tarefas de classificação, sendo por décadas a representação padrão do Processamento de Linguagem Natural estatístico.

2.4.3 Word2Vec

O *Word2Vec*, proposto por Mikolov et al. (2013), inaugurou a era das representações densas: em vez de vetores esparsos de contagem, cada palavra é mapeada para um vetor contínuo de baixa dimensionalidade (tipicamente 300 posições), aprendido por uma rede neural rasa treinada sobre grandes corpora. O treinamento explora a hipótese distribucional — palavras que ocorrem em contextos semelhantes têm significados semelhantes — por meio de duas arquiteturas: o CBOW (*Continuous Bag of Words*), que prevê uma palavra a partir de seu contexto, e o *Skip-gram*, que faz o inverso. O resultado é um espaço vetorial com estrutura semântica notável, no qual palavras próximas em significado ficam próximas

geometricamente e relações analógicas emergem como operações vetoriais — o exemplo clássico é $\text{vetor}(\textit{“king”}) - \text{vetor}(\textit{“man”}) + \text{vetor}(\textit{“woman”}) \approx \text{vetor}(\textit{“queen”})$.

Para representar um documento inteiro, a estratégia usual é calcular a média dos vetores das palavras que o compõem, produzindo um vetor denso de dimensão fixa. Essa agregação injeta semântica pré-treinada na representação — palavras diferentes com significados próximos contribuem de forma semelhante —, mas dilui a informação em textos longos e, como o vetor de cada palavra é estático, não distingue sentidos diferentes de uma mesma palavra em contextos distintos. A palavra *“bank”*, por exemplo, recebe o mesmo vetor em *“river bank”* (margem de rio) e em *“bank loan”* (empréstimo bancário). Os *Transformers* (Seção 2.3) superam essa limitação ao produzir um vetor diferente para cada ocorrência de uma palavra, calculado a partir das demais palavras da sentença pelo mecanismo de atenção — os chamados *embeddings* contextuais. Essa capacidade, contudo, tem o custo computacional elevado discutido ao longo desta dissertação.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma análise crítica dos trabalhos relacionados à detecção automática de *fake news*, organizada segundo a evolução paradigmática observada na literatura: dos métodos clássicos de Aprendizado de Máquina, passando pelas arquiteturas de *Deep Learning* e *Transformers*, até as abordagens desenvolvidas para o contexto da COVID-19 e, por fim, os trabalhos que aplicam a Computação Hiperdimensional a tarefas de detecção. O capítulo organiza-se em seis seções: Aprendizado de Máquina clássico (Seção 3.1); *Deep Learning* e *Transformers* (Seção 3.2); verificação de fatos e modelagem de evidências (Seção 3.3); detecção de *fake news* no contexto da COVID-19 (Seção 3.4); Computação Hiperdimensional em tarefas de detecção (Seção 3.5); e síntese e posicionamento desta dissertação (Seção 3.6), que inclui um quadro comparativo dos trabalhos revisados.

Diferentemente do Capítulo 2, que apresentou a fundamentação conceitual dos paradigmas envolvidos, este capítulo concentra-se nos achados empíricos e nas escolhas metodológicas dos principais trabalhos da área. Essa revisão crítica é essencial para situar a presente pesquisa no contexto do conhecimento atual, evidenciar as contribuições e as limitações dos trabalhos anteriores, justificar as escolhas metodológicas adotadas e, sobretudo, delimitar a lacuna que a dissertação se propõe a preencher, posicionando ao final sua contribuição original.

3.1 Aprendizado de Máquina Clássico

Os primeiros avanços significativos na detecção automática de *fake news* ocorreram com a utilização de algoritmos clássicos de Aprendizado de Máquina aplicados ao Processamento de Linguagem Natural (PLN). Esses modelos fundamentam-se na extração manual de características textuais, empregando técnicas estatísticas como *Bag of Words* (BoW), TF-IDF e análise de *N-Grams* para representar documentos em espaços vetoriais, sobre os quais operam algoritmos como a *Support Vector Machine* (SVM), a *Logistic Regression* (LR) e o *Random Forest* (RF).

Ahmed, Traore e Saad (2017), utilizando o ISOT Fake News Dataset que eles próprios introduziram, demonstraram que a SVM combinada com características baseadas em *N-Grams* (uni-, bi- e trigramas) e TF-IDF alcançou acurácias superiores a 90% na distinção entre notícias verdadeiras e falsas. Esse algoritmo constrói hiperplanos ótimos que

maximizam a separação entre as classes em espaços de alta dimensionalidade e destaca-se pela elevada capacidade de generalização, sobretudo em conjuntos de dados textuais linearmente separáveis; seu custo computacional, entretanto, cresce de forma acentuada em grandes volumes de dados, tornando-o menos eficiente em cenários de *Big Data*.

A *Logistic Regression*, por sua vez, teve ampla utilização em estudos de *fake news* devido à simplicidade matemática, à interpretabilidade e à eficiência em ambientes esparsos. Shu et al. (2017) destacam que modelos lineares apresentam desempenho competitivo quando combinados com representações TF-IDF, sobretudo na classificação binária, calculando as probabilidades associadas às classes por meio da função sigmoide e permitindo identificar quais termos mais influenciam a decisão; sua principal limitação reside na dificuldade de capturar relações semânticas complexas e padrões não lineares.

O *Random Forest* introduziu abordagens baseadas em *ensemble learning* na detecção de desinformação, construindo múltiplas árvores de decisão independentes e combinando-as por votação majoritária. Kaliyar, Goswami e Narang (2021) demonstraram sua elevada robustez em cenários ruidosos e sua capacidade de capturar interações não lineares entre palavras e padrões associados ao sensacionalismo digital, ao custo de maior consumo de memória e de inferência mais onerosa que a dos classificadores lineares.

Em conjunto, esses trabalhos estabeleceram as bases da detecção automática de *fake news*, evidenciando que características linguísticas, frequência de termos e padrões estatísticos permitem distinguir conteúdo falso de notícias legítimas, conforme sistematizado em *surveys* recentes da área (ALGHAMDI; LUO; LIN, 2024; SINGH et al., 2024). Tais abordagens, porém, mostram-se limitadas quanto à compreensão contextual, à ironia, ao sarcasmo e à dependência semântica de longo alcance — limitações que motivaram a adoção de arquiteturas de *Deep Learning*.

3.2 *Deep Learning e Transformers*

As arquiteturas de *Deep Learning* distinguem-se dos métodos clássicos por realizarem automaticamente a extração de características, reduzindo a dependência de engenharia manual de atributos. As *Convolutional Neural Networks* (CNN), adaptadas ao PLN a partir do trabalho de Kim (2014), demonstraram elevada capacidade de identificar padrões locais em sequências textuais, capturando combinações de palavras e estruturas sintáticas associadas à desinformação. Wang (2017) introduziu o LIAR Dataset justamente com arquiteturas híbridas baseadas em CNN, fundindo o vetor de características do texto a um vetor de metadados

sobre o orador (partido, histórico de mentiras, cargo, local da fala) e demonstrando que a incorporação de metadados melhora significativamente a acurácia — estabelecendo um paradigma no qual o contexto estruturado é tão relevante quanto a semântica.

Para superar as limitações das CNNs na modelagem de dependências de longo alcance, surgiram as arquiteturas recorrentes baseadas em *Long Short-Term Memory* (LSTM). Ruchansky, Seo e Liu (2017) demonstraram, com o modelo CSI, que abordagens híbridas baseadas em LSTM aprimoram a detecção ao considerar não apenas o conteúdo textual, mas também padrões temporais e a propagação social das notícias, ainda que ao custo de elevado esforço computacional e de difícil paralelização durante o treinamento.

O avanço mais significativo veio com a arquitetura *Transformer*, proposta por Vaswani et al. (2017) e cujos fundamentos foram apresentados no Capítulo 2. Baseados no mecanismo de *Self-Attention*, os *Transformers* viabilizaram o processamento paralelo de sequências e a captura eficiente de dependências semânticas globais, dando origem ao BERT (DEVLIN et al., 2019) — que se consolidou como principal referência para classificação textual — e, posteriormente, ao RoBERTa (LIU et al., 2019), que elevou ainda mais o desempenho em tarefas de classificação e verificação de fatos.

Apesar da elevada acurácia, esses modelos impõem custo computacional, consumo energético e necessidade de *hardware* especializado consideráveis. Strubell, Ganesh e McCallum (2019) mostram que, em seu caso mais extremo — o treinamento com busca de arquitetura neural —, um modelo *Transformer* pode emitir CO₂ em volume comparável ao de cinco automóveis ao longo de sua vida útil, o que motivou a busca por alternativas mais eficientes, entre elas a Computação Hiperdimensional. Em fronteira mais recente, Hu et al. (2024) investigaram o uso de *Large Language Models* (como GPT-3.5 e GPT-4) na detecção de *fake news* sobre o LIAR Dataset, evidenciando que arquiteturas generativas massivas atingem desempenho competitivo em cenários *few-shot*, porém com latência e consumo energético proibitivos para aplicações em tempo real ou em *edge computing*.

Em contraponto à centralização, surgiram propostas de *Federated Learning*. Jayakody, Mohammad e Halgamuge (2022) propuseram um modelo descentralizado de *Deep Learning* aplicado ao conjunto de dados ISOT, permitindo treinar detectores sem centralizar os dados dos usuários; a solução resolve questões de privacidade, mas ainda depende de redes neurais pesadas em cada nó da rede.

3.3 Verificação de Fatos e Modelagem de Evidências

A linha de pesquisa inaugurada pelo LIAR evoluiu para a modelagem de justificativa (*Justification Modeling*): a proposta de Alhindi, Petridis e Muresan (2018), denominada LIAR-PLUS, incorporou ao modelo a capacidade de identificar não apenas se uma notícia é falsa, mas qual evidência textual justifica esse veredito, aumentando a transparência e a complexidade do processamento.

Nessa mesma direção, a tarefa de verificação factual consolidou-se como formulação mais exigente do problema. Thorne et al. (2018) introduziram o FEVER (*Fact Extraction and VERification*), *benchmark* composto por 185.445 afirmações curtas derivadas de artigos da Wikipédia e rotuladas manualmente em três categorias — *supported*, *refuted* e *not enough info*. Diferentemente dos corpora de *fake news* construídos a partir de artigos jornalísticos completos, nos quais marcadores estilísticos e sensacionalistas fornecem forte sinal discriminativo, o FEVER exige julgar a veracidade de sentenças isoladas cuja verdade não é inferível do próprio enunciado: a formulação original prevê a recuperação de sentenças de evidência no corpus da Wikipédia antes da decisão. O FEVER representa, assim, um cenário de teste substancialmente mais severo para classificadores que operam apenas sobre padrões léxico-sintáticos do texto de entrada, sem acesso a bases de conhecimento externas — condição que caracteriza tanto os modelos clássicos quanto o paradigma hiperdimensional investigado nesta dissertação.

3.4 Detecção de *Fake News* no Contexto da COVID-19

O ferramental de *Deep Learning* e *Transformers* foi intensamente mobilizado durante a pandemia de COVID-19, que provocou, além da crise sanitária, uma crise informacional sem precedentes. O termo “infodemia”, adotado pela Organização Mundial da Saúde, descreve o excesso simultâneo de informações verdadeiras e falsas circulando nas redes digitais e tornou a detecção automática de *fake news* sobre a COVID-19 uma das principais frentes de pesquisa em Inteligência Artificial aplicada ao combate à desinformação.

Os estudos iniciais concentraram-se em modelos baseados em *Transformers* para a análise textual de postagens em redes sociais. Gundapu e Mamidi (2021) aplicaram o BERT à detecção de *fake news* sobre a COVID-19, mostrando que arquiteturas bidirecionais distinguem, por exemplo, o uso irônico de um termo médico de seu uso factual, com *F1-Scores* superiores a 90%, enquanto Patwa et al. (2021) consolidaram a área ao publicar um

benchmark padronizado de notícias falsas sobre a pandemia, que se tornou referência para a comunidade.

Para além das abordagens puramente supervisionadas sobre o texto, trabalhos passaram a integrar fontes adicionais de informação ao conteúdo textual. Paka et al. (2022) propuseram o Cross-SEAN, modelo de atenção neural semi-supervisionado com arquitetura *cross-stitch*, que combina o texto das postagens, características dos usuários e conhecimento externo relevante; embora eficaz, o modelo apresenta alta complexidade computacional e depende de grande volume de dados auxiliares para o treinamento.

A literatura sobre COVID-19 também evidenciou o desbalanceamento de classes como desafio recorrente: em muitos cenários, as notícias reais constituem a maioria dos exemplos, enquanto o conteúdo falso figura como evento raro, o que leva os classificadores tradicionais a favorecer a classe majoritária. Esse problema impulsionou o interesse por métodos adaptativos e por estratégias de aprendizado contínuo capazes de preservar a sensibilidade à detecção da classe minoritária.

3.5 Computação Hiperdimensional em Tarefas de Detecção

Paralelamente a essa evolução, a Computação Hiperdimensional consolidou-se como paradigma de aprendizado eficiente em domínios diversos. Além dos avanços arquiteturais discutidos na Seção 2.2.6, a literatura registra aplicações bem-sucedidas em reconhecimento de idiomas e classificação de texto por codificação de *n*-gramas (JOSHI; HALSETH; KANERVA, 2017), em processamento de sinais biomédicos e robótica (KLEYKO et al., 2023b) e em sistemas de aprendizado embarcados em dispositivos de borda (ZOU et al., 2021), o que evidencia a maturidade do paradigma em tarefas de classificação supervisionada sob restrição de recursos.

Embora o paradigma já tenha sido aplicado à classificação de textos em tarefas como o reconhecimento de idiomas, não foram identificados, na revisão conduzida para esta dissertação, trabalhos que apliquem a Computação Hiperdimensional especificamente à detecção de *fake news* ou a tarefas equivalentes de classificação de veracidade sobre dados textuais. Uma explicação plausível para essa lacuna é a trajetória do próprio paradigma: as variantes revisadas foram propostas majoritariamente em conferências de automação de projeto eletrônico e de *hardware* — como DAC, DATE, FCCM e ICCAD —, com ênfase em eficiência de implementação, enquanto a detecção de *fake news* consolidou-se na comunidade de Processamento de Linguagem Natural, dominada por representações aprendidas. A

literatura mais próxima do problema situa-se no domínio da detecção de anomalias (*outlier detection*) sobre dados tabulares — tarefa estruturalmente análoga, por também buscar identificar instâncias que desviam do comportamento esperado, frequentemente sob forte desbalanceamento entre a classe normal e a classe anômala. Dois trabalhos destacam-se nessa vertente.

O ODHD (WANG; JIAO; HU, 2022) e o IFODHD (XU et al., 2025), cujo funcionamento foi descrito nas Seções 2.2.6.6 e 2.2.6.7, são os trabalhos mais representativos dessa vertente. Do ponto de vista empírico, o ODHD, avaliado em *benchmarks* tabulares, superou métodos consolidados de detecção de *outliers* preservando a eficiência característica do paradigma, e o IFODHD reportou ganhos de acurácia e redução de falsos positivos frente a métodos HDC convencionais ao ponderar os atributos mais informativos antes da codificação, sem comprometer o baixo custo computacional.

Para os propósitos desta dissertação, esses trabalhos são duplamente relevantes. Por um lado, demonstram a viabilidade da HDC em tarefas de detecção nas quais uma das classes é rara ou anômala — situação análoga à dos cenários de infodemia, em que o conteúdo falso é minoritário. Por outro, evidenciam que a literatura existente opera sobre dados tabulares de dimensionalidade moderada, deixando em aberto a questão de se — e sob quais condições — o paradigma hiperdimensional é eficaz sobre dados textuais esparsos e de vocabulário extenso, como os que caracterizam a detecção de *fake news*.

3.6 Síntese e Posicionamento desta Dissertação

A revisão da literatura evidencia a lacuna que motiva esta dissertação: a escalada de complexidade computacional. De *N-Grams* estatísticos a LLMs e redes neurais em grafos, o custo da detecção cresceu ordens de magnitude, sem que houvesse investigação sistemática de paradigmas alternativos capazes de oferecer *trade-offs* mais favoráveis para cenários de *Big Data* e *edge computing*. Análises críticas recentes da área (HAMED; AB AZIZ; YAAKUB, 2023) reforçam que as limitações de representação de características e o custo de fusão de dados permanecem desafios centrais não resolvidos pela literatura dominante.

Embora a literatura em HDC seja madura na classificação de sinais biomédicos, no sensoriamento e na linguagem natural genérica (KANERVA, 2009; IMANI et al., 2019a, 2019b; ZOU et al., 2021), os trabalhos existentes em tarefas de detecção concentram-se em anomalias sobre dados tabulares (WANG; JIAO; HU, 2022; XU et al., 2025), não tendo sido identificados estudos que avaliem sistematicamente o paradigma hiperdimensional para a

detecção de *fake news* em dados textuais, tampouco em comparação direta com o estado da arte.

O Quadro 2 sintetiza os trabalhos revisados neste capítulo, indicando o paradigma adotado, os dados utilizados e a principal contribuição de cada um, e posiciona a presente dissertação em relação a eles.

Quadro 2 – Síntese dos Trabalhos Relacionados

Trabalho	Paradigma	Dados	Contribuição principal
Ahmed, Traore e Saad (2017)	Aprendizado de Máquina clássico (SVM, <i>N-Grams</i> , TF-IDF)	ISOT	Acurácia superior a 90%; introdução do ISOT
Shu et al. (2017)	Aprendizado de Máquina clássico (modelos lineares, TF-IDF)	Levantamento (<i>survey</i>)	Competitividade de modelos lineares na classificação binária
Wang (2017)	<i>Deep Learning</i> (CNN híbrida com metadados)	LIAR	Metadados do orador elevam a acurácia; introdução do LIAR
Ruchansky, Seo e Liu (2017)	<i>Deep Learning</i> (LSTM híbrida — CSI)	Texto, padrões temporais e propagação social	Incorporação da dinâmica de propagação, com alto custo de treinamento
Alhindi, Petridis e Muresan (2018)	Modelagem de justificativa	LIAR-PLUS	Identificação da evidência que justifica o veredito
Thorne et al. (2018)	Verificação de fatos com recuperação de evidências	FEVER	<i>Benchmark</i> de 185.445 afirmações rotuladas contra a Wikipédia
Gundapu e Mamidi (2021)	<i>Transformers</i> (BERT)	COVID-19	<i>F1-Score</i> superior a 90% em postagens sobre a pandemia
Patwa et al. (2021)	Conjunto de dados	COVID-19	<i>Benchmark</i> padronizado de notícias falsas sobre a pandemia
Paka et al. (2022)	Atenção neural semi-supervisionada (Cross-SEAN)	COVID-19, usuários e conhecimento externo	Integração de fontes auxiliares, com alta complexidade computacional
Jayakody, Mohammad e Halgamuge (2022)	<i>Deep Learning</i> descentralizado (<i>Federated Learning</i>)	ISOT	Treinamento sem centralizar dados, com redes pesadas em cada nó
Wang, Jiao e Hu (2022)	HDC de classe única (ODHD)	<i>Benchmarks</i> tabulares	Detecção de <i>outliers</i> com seleção automática de limiar
Xu et al. (2025)	HDC com seleção de características (IFODHD)	<i>Benchmarks</i> tabulares	Redução de falsos positivos pela ponderação de atributos
Esta dissertação	HDC (cinco variantes) × Aprendizado de Máquina clássico × <i>Transformers</i>	ISOT, COVID-19 e FEVER	Avaliação sistemática da HDC na detecção de <i>fake news</i> textual, com foco no <i>trade-off</i> acurácia × eficiência energética

Fonte: Elaborado pelo autor (2026).

Esta dissertação posiciona-se, portanto, no preenchimento dessa lacuna, diferenciando-se dos trabalhos revisados em três aspectos. Primeiro, enquanto a literatura de HDC em tarefas de detecção se restringe a anomalias em dados tabulares, este trabalho investiga o paradigma sobre dados textuais de vocabulário extenso, no problema da detecção de *fake news*. Segundo, diferentemente dos estudos que avaliam cada paradigma isoladamente, realiza-se aqui uma avaliação comparativa sistemática da HDC frente a abordagens consolidadas e ao estado da arte, sob um protocolo experimental unificado. Terceiro, em contraste com a ênfase quase exclusiva em acurácia observada na literatura de *fake news*, o *trade-off* entre desempenho preditivo e eficiência energética é tratado como objeto central de investigação, caracterizando-se empiricamente os cenários em que a HDC oferece vantagem prática, sem reivindicações de superioridade absoluta. Os detalhes das escolhas experimentais que operacionalizam essa investigação são apresentados no Capítulo 4.

4 METODOLOGIA

Este capítulo descreve a metodologia adotada para validar a hipótese de que a Computação Hiperdimensional (HDC) oferece um *trade-off* favorável entre acurácia e eficiência energética na detecção de *fake news*. A organização segue as etapas canônicas de um trabalho experimental de Aprendizado de Máquina: após o delineamento geral (Seção 4.1), apresentam-se a coleta de dados (Seção 4.2), o pré-processamento (Seção 4.3), a extração de *features* (Seção 4.4), os modelos de classificação e seus hiperparâmetros (Seção 4.5), as métricas de avaliação (Seção 4.6) e, por fim, o procedimento de estimação do consumo energético (Seção 4.7), que operacionaliza a dimensão de eficiência central a esta dissertação.

4.1 Delineamento Metodológico

A pesquisa adota uma abordagem quantitativa-experimental. A natureza quantitativa justifica-se pela necessidade de mensurar objetivamente o desempenho dos classificadores através de métricas padronizadas (Acurácia, Precisão, Revocação, *F1-Score* e área sob a curva ROC — AUC), complementadas pelo instrumento diagnóstico da matriz de confusão, permitindo a comparação direta com o Estado da Arte revisado na literatura. O caráter experimental reside na manipulação controlada das variáveis independentes, especialmente a arquitetura do modelo, a representação textual e o tipo de conjunto de dados utilizado, com o objetivo de observar seus efeitos sobre as variáveis dependentes (eficácia da classificação e custo computacional).

O fluxo metodológico estrutura-se em cinco fases sequenciais: (1) aquisição e consolidação de dados; (2) pré-processamento textual; (3) codificação vetorial (BoW, TF-IDF, *Word2Vec*); (4) treinamento e inferência dos modelos; e (5) avaliação comparativa de eficácia e eficiência energética. Na primeira etapa, realizou-se a unificação dos formatos heterogêneos dos conjuntos de dados ISOT, COVID-19 e FEVER, permitindo a construção de um *pipeline* padronizado de processamento textual.

O sistema foi implementado seguindo um *pipeline* modular, otimizado para processamento em lote, essencial para lidar com o volume massivo do conjunto de dados FEVER (superior a 100.000 instâncias). A implementação foi realizada em Python 3.10, utilizando *scikit-learn* para os modelos de referência clássicos, a biblioteca *torchhd* (HEDDES et al., 2023) para os modelos HDC, e *HuggingFace Transformers* para BERT e RoBERTa.

Para garantir que as métricas de eficiência sejam válidas e comparáveis, todos os experimentos foram executados em ambiente computacional padronizado: Google Colab, em sua versão gratuita, com instância equipada com GPU NVIDIA Tesla T4 (16 GB VRAM, TDP 70W), 12 GB de RAM e processador Intel Xeon (2 vCPUs). Os modelos HDC foram executados majoritariamente em modo GPU para acelerar as operações vetoriais, e os *Transformers* utilizaram a mesma GPU para treinamento e inferência, garantindo comparação justa em termos de *wall-clock time* e consumo energético estimado.

4.2 Coleta de Dados

Esta seção descreve os conjuntos de dados que sustentam a avaliação experimental: a justificativa de sua seleção, a caracterização de cada corpus (volumes, classes e particularidades), o processo de ingestão e normalização — incluindo a preparação da versão binarizada do FEVER — e o particionamento dos dados em conjuntos de treinamento e teste.

A escolha dos três conjuntos de dados — ISOT, COVID-19 e FEVER — não foi arbitrária, mas desenhada metodologicamente para isolar e estressar variáveis críticas do problema de classificação de texto:

- **ISOT Fake News Dataset (Padrão de Referência em Conteúdo):** selecionado por ser um *benchmark* consolidado na literatura para análise de conteúdo em textos longos e estruturados — artigos jornalísticos completos, formato distinto do de parte relevante da desinformação em redes sociais, que circula em textos curtos, como postagens e manchetes, representados nesta dissertação sobretudo pelo conjunto COVID-19. Compilado originalmente para validar a eficácia da análise de *N-Grams* combinada com Aprendizado de Máquina (AHMED; TRAORE; SAAD, 2017), possui distribuição de classes balanceada. Sua utilização permite avaliar a capacidade intrínseca do HDC de aprender padrões sintáticos e léxicos sem o viés estatístico introduzido por classes desproporcionais, servindo como “grupo de controle” essencial.
- **COVID-19 Fake News Dataset (Desafio do Desbalanceamento):** a inclusão deste conjunto de dados (PATWA et al., 2021) responde à necessidade de testar o modelo em contextos de alta volatilidade e especificidade técnica. Como apontado em *surveys* abrangentes, conjuntos de dados focados na COVID-19 são cruciais para entender como modelos computacionais lidam com a infodemia (SHAHI; DIRKSON; MAJCHRZAK, 2022). A justificativa para seu uso reside no desafio estatístico: o

conjunto apresenta desbalanceamento de aproximadamente 3:1, testando se o HDC consegue evitar o viés de superclassificar a classe majoritária.

- **FEVER Dataset (Teste de Escalabilidade e Contexto):** o FEVER (*Fact Extraction and VERification*) é incluído como o teste de estresse máximo do sistema proposto. Introduzido como um *benchmark* de larga escala composto por *claims* factuais curtos, derivados de artigos da Wikipédia e rotulados manualmente (THORNE et al., 2018), representa o cenário de “pior caso”: pouco contexto textual disponível por instância e alta demanda por conhecimento externo. Nesta dissertação, foi utilizada a partição de treinamento do FEVER em versão binarizada, totalizando 109.809 instâncias rotuladas, conforme descrito em detalhe na Seção 4.2.1. Cabe distinguir, contudo, que o FEVER foi concebido como *benchmark* de verificação de fatos (*fact-checking*) — tarefa em que a veracidade de uma afirmação é julgada contra evidências recuperadas de uma base de conhecimento —, e não de classificação de notícias pelo conteúdo textual; seu uso nesta dissertação, em configuração que classifica apenas o texto do *claim*, é justificado e delimitado na Seção 4.2.2.

O Quadro 3 sintetiza as características dos três *corpora* empregados na avaliação experimental.

Quadro 3 – Características dos Conjuntos de Dados Utilizados

Conjunto	Tipo de texto e origem	Instâncias	Real / Fake	Razão	Papel no estudo
ISOT	Artigos jornalísticos completos: Reuters (real) e fontes classificadas como não confiáveis por organizações de <i>fact-checking</i> (<i>fake</i>) (AHMED; TRAORE; SAAD, 2017)	44.267	21.416 / 22.851	$\approx 1:1$	Controle (classes balanceadas)
COVID-19	Postagens de redes sociais e notícias curtas sobre a pandemia, vacinas e tratamentos, na linhagem de Patwa et al. (2021)	5.975	4.532 / 1.443	$\approx 3:1$	Robustez ao desbalanceamento
FEVER (binarizado)	Afirmações (<i>claims</i>) curtas derivadas da Wikipédia; SUPPORTS $\rightarrow$ real e REFUTES $\rightarrow$ <i>fake</i> (THORNE et al., 2018)	109.809	80.034 / 29.775	$\approx 2,7:1$	Escalabilidade

Fonte: Elaborado pelo autor (2026).

Cabe registrar que a versão do COVID-19 Fake News Dataset utilizada neste trabalho difere do *release* original do *benchmark* de Patwa et al. (2021), aproximadamente balanceado e com 10.700 postagens: trata-se de uma versão consolidada com 5.975 instâncias e forte

desbalanceamento, descrita na Seção 4.2.1. No FEVER, os textos são muito curtos e a veracidade depende fortemente de conhecimento externo, aspecto discutido na Seção 4.2.2.

4.2.1 Ingestão e Normalização dos Dados

O módulo de ingestão é responsável por carregar os arquivos brutos e normalizar as estruturas de dados. Diferentes fontes apresentam formatações distintas:

- **ISOT:** estruturado em dois arquivos CSV separados (*True.csv* e *Fake.csv*), exigindo fusão e rotulagem binária (0 para Real, 1 para *Fake*). Total: 44.267 instâncias após remoção de duplicatas.
- **COVID-19:** conjunto de dados na linhagem do *benchmark* de Patwa et al. (2021), em versão consolidada com 5.975 instâncias agregando postagens de Twitter e bases de verificadores de fatos, cuja distribuição de classes difere do *release* original (ver Seção 4.2). Exigiu tratamento de caracteres especiais, *hashtags* e menções típicas de redes sociais.
- **FEVER (versão binarizada):** o FEVER original (THORNE et al., 2018) contém 185.445 *claims*, dos quais 145.449 compõem a partição de treinamento (80.035 SUPPORTS, 29.775 REFUTES e 35.639 NOT ENOUGH INFO). Para esta dissertação foi preparada uma versão binária por meio do seguinte processo: (i) seleção da partição de treinamento, a única distribuída com rótulos públicos em volume compatível com o teste de escalabilidade; (ii) descarte da classe NOT ENOUGH INFO, que não possui equivalente na formulação binária real/*fake*; (iii) mapeamento dos rótulos SUPPORTS → “real” (classe 0) e REFUTES → “*fake*” (classe 1) — aproximação discutida na Seção 4.2.2; (iv) normalização textual e remoção de duplicatas exatas por *hash* do conteúdo. O resultado é um conjunto de 109.809 *claims* com distribuição 80.034 reais / 29.775 *fake* (razão aproximada de 2,7:1).

A versão binarizada do FEVER, embora se afaste da formulação original da tarefa — que inclui a recuperação de sentenças de evidência na Wikipédia —, preserva a natureza essencial dos dados (*claims* factuais curtos com veracidade anotada manualmente) e foi escolhida especificamente para permitir o teste de escalabilidade descrito no objetivo OE3, condição que corpora menores, como o LIAR (12,8 mil instâncias; WANG, 2017), não permitiriam explorar adequadamente.

4.2.2 Considerações Metodológicas sobre o FEVER Binarizado

A utilização do FEVER (THORNE et al., 2018) em formulação binária e sem a etapa de recuperação de evidências suscita quatro pontos metodológicos que merecem registro explícito.

(i) Descarte da classe NOT ENOUGH INFO e deduplicação. A formulação original do FEVER é ternária, e a classe NOT ENOUGH INFO (35.639 *claims* da partição de treinamento) não possui equivalente direto na tarefa binária *real/fake* adotada nesta dissertação; seu descarte é prática recorrente quando o FEVER é empregado em configuração de classificação binária, mas implica que os resultados aqui reportados não são comparáveis aos da tarefa ternária original. Complementarmente, aplicou-se remoção de duplicatas exatas por *hash* SHA-256 do conteúdo textual normalizado (*lowercase*, remoção de pontuação redundante e espaços excedentes), responsável pela diferença de uma instância em relação à contagem oficial da partição (80.035 SUPPORTS). A divisão treino/teste foi realizada após a deduplicação e de forma estratificada com *random_state* fixo.

(ii) Classificação apenas do *claim*, sem recuperação de evidências. A tarefa completa do FEVER exige recuperar sentenças de evidência da Wikipédia e verificar o *claim* contra elas; a métrica oficial do *shared task* (*FEVER Score*) condiciona o acerto do rótulo à apresentação de evidência correta. Nesta dissertação, os modelos operam exclusivamente sobre o texto do *claim* (configuração *claim-only*), sem acesso à base de evidências. Trata-se de uma formulação deliberadamente mais exigente no eixo semântico — a veracidade de muitos *claims* não é inferível apenas do texto — e alinhada ao objetivo da pesquisa, que avalia classificadores de conteúdo textual puro. Os resultados de *F1-Score* reportados (até 53,6% nas variantes HDC; 69–70% nos *Transformers*) devem ser interpretados sob essa lente e não são comparáveis aos de sistemas completos do FEVER *shared task*.

(iii) Natureza dos *claims* e generalização de domínio. Os *claims* do FEVER são derivados de artigos da Wikipédia por anotadores humanos e cobrem entidades e fatos de domínio geral (pessoas, obras, lugares, eventos), diferentemente das declarações políticas espontâneas de corpora como o LIAR (WANG, 2017). Essa característica torna o corpus adequado ao teste de escalabilidade e à avaliação da dependência de conhecimento externo, mas implica que o desempenho observado não se transfere automaticamente para o domínio político-jornalístico; a avaliação cross-domain permanece como trabalho futuro.

(iv) Correspondência entre os rótulos do FEVER e a classificação binária. No FEVER, os rótulos SUPPORTS e REFUTES indicam se a afirmação é sustentada ou

contradita pelas evidências da Wikipédia, e não se ela é verdadeira ou falsa em termos absolutos. O mapeamento SUPPORTS $\rightarrow$ real e REFUTES $\rightarrow$ *fake* adotado nesta dissertação é, portanto, uma aproximação operacional, que assume a Wikipédia como fonte de referência de veracidade; os rótulos “real” e “*fake*” no FEVER devem ser lidos como “sustentado” e “refutado” por essa fonte. Soma-se a isso a diferença de natureza entre as tarefas: o FEVER foi concebido para verificação de fatos (*fact-checking*), e não para detecção de *fake news* pelo conteúdo, de modo que os resultados obtidos sobre ele medem a capacidade dos modelos de explorar regularidades textuais dos *claims*, e não de verificar sua veracidade.

4.2.3 Particionamento dos Conjuntos de Treinamento e Teste

Nenhum dos três conjuntos de dados fornece divisão oficial única compatível com o protocolo unificado desta dissertação — o ISOT e o COVID-19 são distribuídos como corpora íntegros, e o FEVER binarizado, por construção, deriva exclusivamente da partição de treinamento original do *shared task*. Assim, para garantir a validade estatística e a comparabilidade entre os cenários, a separação entre treinamento e teste seguiu o mesmo protocolo (proporção 70/30 e semente fixa) em todos os corpora e modelos; como o particionamento é executado de forma independente em cada *pipeline* de codificação, a composição exata das partições pode apresentar variações residuais entre *pipelines*, registradas no Apêndice A. Não foi reservado conjunto de validação separado, e todas as avaliações foram realizadas no conjunto de teste: os hiperparâmetros dos modelos clássicos foram mantidos nos valores padrão da biblioteca, os dos *Transformers* seguiram a configuração usual da literatura, restrita pelo *hardware* disponível (Seção 4.5.2), e, para o HDC, a dimensionalidade e a codificação adotadas na comparação final foram definidas a partir da grade completa de resultados reportada nos Experimentos 1 e 2 (Seções 5.3 e 5.4) — escolha registrada como limitação na Seção 5.8.

Os conjuntos de dados foram divididos seguindo a proporção padrão de 70/30 com estratificação por classe:

- **70% para treinamento:** utilizados para construir os protótipos de classe (HDC) ou ajustar os pesos (ML clássico e *Transformers*).
- **30% para teste:** utilizados para avaliar a generalização do modelo em dados nunca vistos (*hold-out set*).

No caso do FEVER binarizado, dado seu volume de 109.809 instâncias, essa divisão resultou em um conjunto de teste de 32.943 amostras, garantindo alta confiança estatística.

4.3 Pré-processamento e Tratamento de Dados

A qualidade da representação textual depende diretamente da limpeza do corpus. A literatura indica que o ruído em dados não estruturados pode degradar a ortogonalidade dos hipervetores em HDC e introduzir vieses nos demais modelos. Aplicou-se, portanto, um protocolo rigoroso de pré-processamento, uniforme para todos os modelos:

- **Remoção de artefatos digitais:** URLs, *tags* HTML, menções de usuários (@) e *hashtags* foram removidas, pois não contribuem para o conteúdo semântico da veracidade.
- **Padronização de caixa:** todo o texto foi convertido para minúsculas (*lowercase*) para evitar duplicação de vetores para a mesma palavra.
- **Remoção de stopwords:** palavras funcionais de alta frequência (artigos, preposições, conjunções) listadas no NLTK *English stopwords* foram filtradas, reduzindo ruído nas operações de *bundling*.
- **Stemming:** aplicou-se o algoritmo *Porter Stemmer* para reduzir palavras ao seu radical invariante (e.g., “*corruption*”, “*corrupt*”, “*corrupted*” → “*corrupt*”). Esta técnica oferece duas vantagens para o HDC: densidade semântica (agrupa variações morfológicas no mesmo hipervetor) e eficiência de memória (reduz o tamanho do dicionário de *Item Memory*).

A aplicação do mesmo protocolo aos *Transformers* foi uma decisão deliberada, adotada para que todos os modelos recebessem exatamente a mesma informação textual, isolando o efeito da arquitetura de classificação. Reconhece-se, contudo, que ela difere da prática usual com esses modelos, pré-treinados sobre texto natural e tipicamente ajustados sobre o texto bruto: o *stemming* produz radicais que o tokenizador *subword* fragmenta, e a lista de *stopwords* do NLTK inclui termos de negação (como “*not*”, “*no*” e “*nor*”), cuja remoção pode alterar o sentido de afirmações curtas — efeito que atinge todos os modelos, mas é particularmente relevante no FEVER. Como a tendência é de que o protocolo tenha prejudicado sobretudo os *Transformers*, os resultados do BERT e do RoBERTa reportados no Capítulo 5 devem ser lidos como uma estimativa conservadora de sua eficácia; a superioridade desses modelos em todos os cenários não é, portanto, enfraquecida por essa escolha. No eixo da eficiência, a remoção de *stopwords* encurta as sequências de entrada, o que tende a reduzir o tempo e a energia dos *Transformers*; também nesse caso o efeito é conservador em relação à vantagem energética atribuída ao HDC.

4.4 Extração de *Features*

Como discutido na Seção 2.4, algoritmos de aprendizado de máquina não operam sobre texto bruto: é necessário converter cada documento em uma representação numérica que preserve as propriedades relevantes para a classificação. Esta seção apresenta as representações textuais adotadas nesta dissertação e, na sequência, a codificação hiperdimensional que projeta essas representações no hiperespaço dos modelos HDC.

4.4.1 Representações Textuais Adotadas

Os experimentos avaliaram três representações textuais distintas, escolhidas para cobrir o espectro entre representações frequenciais esparsas e densas:

- ***Bag of Words (BoW)***: representação frequencial esparsa baseada em contagem de *tokens*. Vocabulário limitado a 10.000 termos mais frequentes para limitar a dimensionalidade.
- ***TF-IDF***: Pondera os *tokens* pela sua relevância no documento e raridade no corpus. Vocabulário também limitado a 10.000 termos.
- ***Word2Vec (W2V)***: *embeddings* densos pré-treinados de 300 dimensões (modelo GoogleNews-vectors-negative300; MIKOLOV et al., 2013). O vetor do documento é a média dos vetores das palavras presentes. Como os vetores foram obtidos a partir do texto já submetido ao pré-processamento da Seção 4.3, incluindo o *stemming*, os radicais que não correspondem a palavras do vocabulário do modelo pré-treinado — por exemplo, “*compani*”, derivado de “*company*” — não encontram vetor correspondente e deixam de contribuir para a representação do documento (limitação discutida na Seção 5.8).

A escolha de avaliar BoW, TF-IDF e *Word2Vec* não é redundante: cada representação testa uma hipótese distinta sobre como o HDC absorve informação. BoW avalia se a mera contagem é suficiente; TF-IDF avalia se a ponderação estatística melhora a separabilidade; *Word2Vec* avalia se a injeção de semântica densa pré-treinada melhora resultados quando combinada à álgebra hiperdimensional. Os modelos *Transformer* (BERT/RoBERTa) utilizam seu próprio tokenizador *subword* nativo — aplicado ao texto já pré-processado conforme a Seção 4.3 —, dispensando representações externas. Não foram avaliados, como entrada para o HDC, *embeddings* contextuais extraídos de *Transformers* — por exemplo, a representação do *token* [CLS] de um BERT congelado —, que combinariam a semântica contextual desses

modelos com a álgebra hiperdimensional; essa combinação é registrada como trabalho futuro (Seção 6.5).

4.4.2 Codificação Hiperdimensional (*Encoding*)

Para os modelos HDC, cada representação é então projetada para o hiperespaço de dimensão D — avaliado em duas configurações, $D = 1.000$ e $D = 3.000$ — conforme protocolo abaixo.

Em todas as variantes, os hipervetores de base são gerados aleatoriamente pela própria biblioteca no momento da instanciação do classificador, a partir do número de atributos do vetor de entrada e da dimensionalidade D escolhida. Foram avaliadas duas dimensões: $D = 1.000$ e $D = 3.000$. Esta alta dimensionalidade garante que quaisquer dois hipervetores de base escolhidos aleatoriamente sejam aproximadamente ortogonais (distância de Hamming esperada $\approx D/2$), propriedade essencial para evitar colisão de informações (KANERVA, 2009).

A restrição do estudo a essas duas configurações decorre de limitação computacional do ambiente experimental. Embora a literatura de HDC frequentemente adote $D = 10.000$ como valor de referência (KANERVA, 2009), a instância gratuita do Google Colab utilizada — GPU NVIDIA Tesla T4 com 16 GB de VRAM e 12 GB de RAM (Seção 4.1) — não comporta essa configuração para a totalidade da grade experimental. O consumo de memória da Memória de Itens e das matrizes de hipervetores cresce linearmente com D , assim como o custo agregado das operações de *bundling* e de cálculo de similaridade a cada época de treinamento. No conjunto FEVER, com 109.809 instâncias, elevar D de 3.000 para 10.000 multiplicaria por aproximadamente 3,3 vezes tanto a memória requerida quanto o tempo de execução, excedendo os limites de sessão do ambiente e inviabilizando a execução completa da grade (variantes HDC $\times$ três codificações $\times$ três conjuntos de dados). A versão gratuita do Google Colab foi o único ambiente com GPU disponível ao autor durante a pesquisa, sem acesso a planos pagos ou a outra infraestrutura com maior capacidade de memória; além disso, migrar parte da grade para outro *hardware* comprometeria a comparabilidade das medidas de tempo e energia, que pressupõem um ambiente único para todos os modelos. Optou-se, portanto, por avaliar $D = 1.000$ e $D = 3.000$ para todos os modelos HDC, o que preserva a uniformidade e a reprodutibilidade da comparação dentro dos recursos disponíveis. Esta restrição é retomada como limitação do estudo na Seção 5.8 e como direção de trabalho futuro na Seção 6.5.

A projeção das representações textuais para o hiperespaço é realizada pelos codificadores internos da biblioteca torchhd (HEDDES et al., 2023). Cada representação — BoW, TF-IDF ou Word2Vec — produz um vetor de entrada de dimensão n , que é então mapeado para um hipervetor de dimensão D pelo codificador associado a cada variante de classificador. No caso do *Vanilla* HDC e do *AdaptHD*, esse mapeamento segue a codificação baseada em registros (*record-based encoding*), na qual cada atributo i do vetor de entrada é associado por *binding* a um hipervetor de posição aleatório ID_i e ao hipervetor de nível Li correspondente ao seu valor, sendo os pares agregados por *bundling*:

$$H_doc = \sum_i (ID_i \otimes Li)$$

Cabe registrar que esse codificador não é uniforme entre as variantes avaliadas. Enquanto o *Vanilla* HDC e o *AdaptHD* adotam a codificação baseada em registros descrita acima, o *OnlineHD* e o *NeuralHD* empregam projeção aleatória não linear e o *DistHD*, projeção aleatória linear. Nenhuma das variantes utiliza codificação de N-gramas ou operações de permutação, de modo que a ordem sequencial dos termos não é preservada em nenhum dos caminhos avaliados: a informação disponível aos classificadores hiperdimensionais é a mesma fornecida aos modelos de referência de Aprendizado de Máquina clássico, o que torna direta a comparação entre os dois paradigmas conduzida no Capítulo 5.

Essa heterogeneidade constitui uma limitação metodológica que deve ser considerada na leitura do Capítulo 5. As diferenças de desempenho observadas entre as variantes HDC refletem a combinação de dois fatores — a estratégia de atualização dos protótipos e o codificador adotado —, não sendo possível isolar a contribuição de cada um sob o protocolo empregado. A comparação entre o paradigma hiperdimensional e os demais paradigmas permanece válida, uma vez que cada variante é avaliada como sistema completo; o que não se pode atribuir com segurança é a diferença observada entre variantes a uma única causa. O isolamento desses efeitos, mediante fixação de um codificador comum a todas as variantes, é registrado como direção de trabalho futuro na Seção 6.5. Em todos os caminhos, preservam-se as propriedades de quase ortogonalidade discutidas na Seção 2.2.1.

4.5 Modelos de Classificação

Para situar empiricamente o desempenho da Computação Hiperdimensional, esta dissertação avalia, além das cinco variantes de HDC, três famílias de modelos comparativos.

Esta seção descreve e justifica cada família de modelos, bem como os hiperparâmetros adotados.

4.5.1 Modelos de Referência de Aprendizado de Máquina Clássico

A primeira família de comparativos reúne cinco classificadores consagrados. A **Logistic Regression (LR)** é um modelo linear que estima a probabilidade de cada classe por meio da função sigmoide, destacando-se pela interpretabilidade e pela eficiência em representações esparsas. A **Support Vector Machine (SVM)** constrói o hiperplano que maximiza a margem de separação entre as classes, com uso de *kernels* para fronteiras não lineares. O **Random Forest (RF)** é um *ensemble* de árvores de decisão independentes combinadas por votação majoritária, robusto a ruído e capaz de capturar interações não lineares entre atributos. A Árvore de Decisão (*Decision Tree*) particiona recursivamente o espaço de atributos por testes de limiar, oferecendo um classificador não linear de baixo custo e alta interpretabilidade. O *K-Nearest Neighbors* (KNN) classifica cada instância pela classe majoritária entre seus *k* vizinhos mais próximos, servindo como referência baseada puramente em similaridade no espaço vetorial. O papel dos três primeiros na literatura de *fake news* foi discutido no Capítulo 3.

Os modelos de referência *Logistic Regression*, SVM, *Random Forest*, Árvore de Decisão e KNN foram selecionados por três razões: (i) são os classificadores mais utilizados em estudos seminais de *fake news* (AHMED; TRAORE; SAAD, 2017; SHU et al., 2017; KALIYAR; GOSWAMI; NARANG, 2021), permitindo comparabilidade direta; (ii) representam paradigmas distintos — linear (LR), baseado em *kernel* (SVM), em *ensemble* de árvores (RF), em partição recursiva do espaço de atributos (Árvore de Decisão) e baseado em instâncias (KNN) —, oferecendo cobertura ampla das abordagens não-neurais; (iii) operam de forma rápida em CPU, servindo como referência para avaliar quando o ganho de eficiência do HDC justifica sua complexidade adicional em relação a um *pipeline* tradicional. Todos foram executados com hiperparâmetros padrão do scikit-learn.

4.5.2 Transformers: BERT e RoBERTa

BERT e RoBERTa foram selecionados como representantes do Estado da Arte por três razões: (i) BERT é o modelo *Transformer* mais empregado em *fake news detection* desde Gundapu e Mamidi (2021), constituindo *benchmark* canônico; (ii) RoBERTa representa a evolução incremental do BERT com pré-treinamento otimizado, oferecendo um teto de desempenho atualizado; (iii) ambos são executáveis em GPU única (Tesla T4), o que mantém

a comparação justa em termos de infraestrutura. LLMs maiores (GPT-3.5, GPT-4) não foram incluídos pois exigem APIs comerciais ou *clusters* multi-GPU, fugindo do escopo de *hardware* controlado desta dissertação — referência paralela aos resultados de Hu et al. (2024) é discutida no Capítulo 5.

Modelos baseados em grafos, como as *Graph Neural Networks* (GNN), frequentes na literatura que modela a propagação de notícias em redes sociais, também não foram incluídos: os três conjuntos de dados utilizados contêm o conteúdo textual das instâncias, sem informações de propagação ou de usuários que permitam construir os grafos exigidos por essas arquiteturas. A extensão do HDC a dados de propagação é registrada como trabalho futuro (Seção 6.5).

Ambos os modelos, cujas arquiteturas foram apresentadas na Seção 2.3, foram submetidos a *fine-tuning* completo para a tarefa de classificação binária, com os seguintes hiperparâmetros: 2 épocas de treinamento, *batch size* de 16, taxa de aprendizado (*learning rate*) de $2e-5$ com o otimizador AdamW e comprimento máximo de sequência de 512 *tokens*. Utilizaram-se os *checkpoints* pré-treinados bert-base-uncased e roberta-base, e a tokenização *subword* nativa de cada modelo. Adotou-se o ajuste fino completo — em que todos os parâmetros do modelo pré-treinado são atualizados, configuração padrão da biblioteca HuggingFace Transformers para classificação —, e não o uso dos modelos como extratores de *features* congelados seguidos de um classificador leve, por ser a configuração com que esses modelos atingem o estado da arte em classificação textual (DEVLIN et al., 2019), isto é, a que representa o teto de eficácia contra o qual o HDC é comparado. Reconhece-se que a extração de *features* reduziria o custo de treinamento dos *Transformers* e constituiria um ponto intermediário relevante no *trade-off* acurácia $\times$ eficiência; sua avaliação é registrada como trabalho futuro (Seção 6.5). O custo de inferência, por outro lado, seria essencialmente o mesmo nas duas abordagens, pois ambas exigem a passagem de cada texto por toda a rede, de modo que a comparação de inferência apresentada na Seção 5.6 independe dessa escolha.

A definição desse orçamento de treinamento também foi condicionada pela limitação computacional descrita na Seção 4.1. O *batch size* de 16 corresponde ao maior valor estável para sequências de 512 *tokens* nos 16 GB de VRAM da GPU Tesla T4, uma vez que lotes maiores provocam esgotamento de memória durante o ajuste fino. O número de épocas, por sua vez, foi limitado a 2 pelo tempo de execução acumulado: somente o treinamento do BERT nos três conjuntos de dados consumiu aproximadamente duas horas de GPU, valor que se repete para o RoBERTa e que cresceria proporcionalmente com épocas adicionais,

ultrapassando os limites de sessão do ambiente. Adotou-se, portanto, o menor orçamento compatível com a convergência usual do ajuste fino de modelos BERT em classificação binária, que a literatura situa na faixa de 2 a 4 épocas.

4.5.3 Modelos de Computação Hiperdimensional

A terceira família compreende as cinco variantes de HDC descritas em detalhe na Seção 2.2.6: o *Vanilla* HDC, adotado como *baseline* clássico do paradigma, e as quatro variantes adaptativas AdaptHD, OnlineHD, NeuralHD e DistHD. Todas compartilham os hiperparâmetros estruturais — dimensionalidade D avaliada em duas configurações (1.000 e 3.000) e 2 épocas de exposição aos dados — e diferenciam-se pelo mecanismo de atualização dos protótipos e, conforme registrado na Seção 4.4.2, também pelo codificador adotado. Os mecanismos de atualização são: soma direta dos hipervetores de treino, sem correção de erro (*Vanilla* HDC), correção iterativa de erro (AdaptHD), atualização adaptativa por similaridade (OnlineHD), regeneração dinâmica de dimensões (NeuralHD) e codificação dinâmica com decisão top-2 e alta paralelização (DistHD). A inclusão do *Vanilla* HDC permite quantificar o ganho efetivo das estratégias adaptativas sobre a formulação original do paradigma. Os parágrafos seguintes detalham o procedimento de treinamento e inferência comum a todas as variantes.

Esse valor de épocas, comum a todas as variantes, foi igualmente condicionado pela limitação computacional do ambiente (Seções 4.1 e 4.5.2): manter o mesmo orçamento de treinamento para modelos HDC e *Transformers* preserva a comparabilidade das medidas de tempo e energia reportadas no Capítulo 5, evitando que a vantagem energética observada decorra de um número desigual de passagens sobre os dados.

Diferentemente das redes neurais profundas, que exigem algoritmos de otimização iterativos e custosos (como o *backpropagation*), os modelos HDC empregam abordagens algébricas leves para construção dos protótipos de classe. O objetivo do treinamento é construir uma representação vetorial holística para cada classe alvo. Para fins de padronização do protocolo experimental, todas as variantes HDC foram executadas com *epochs* = 2 nesta dissertação — inclusive o OnlineHD que, embora concebido para aprendizado em passagem única, foi exposto ao mesmo número de épocas para garantir comparabilidade direta entre as variantes. No contexto da detecção binária, o sistema constrói dois hipervetores globais, denominados Protótipos de Classe:

- **Protótipo Real (P_Real):** representa o perfil agregado de todas as notícias verídicas do conjunto de treinamento.
- **Protótipo Fake (P_Fake):** representa o perfil agregado de todas as notícias fabricadas ou rumores.

A construção desses protótipos utiliza a operação de *Bundling*. Para cada classe $c \in \{\text{Real}, \text{Fake}\}$, o protótipo P_c é calculado pela soma vetorial limiarizada de todos os N_c hipervetores de treinamento daquela classe:

$$P_c = \text{threshold}(\sum_{i=1}^{N_c} H_{\text{doc},i}), \quad \text{onde } \text{threshold}(x) = 1 \text{ se } x > N_c/2, \text{ senão } 0.$$

Em implementações com vetores binários, a votação majoritária bit a bit é aplicada após a soma. As variantes AdaptHD, OnlineHD, NeuralHD e DistHD modificam este procedimento básico introduzindo correção iterativa (AdaptHD), atualização adaptativa por similaridade (OnlineHD), regeneração de dimensões pouco discriminativas (NeuralHD) e codificação dinâmica com decisão top-2 (DistHD), conforme apresentado na Seção 2.2.6.

A classificação de uma nova instância (notícia de teste) é realizada através de operações de similaridade no hiperespaço:

- **1. Codificação da consulta:** a notícia desconhecida Q passa pelo mesmo *pipeline* de pré-processamento e codificação utilizado no treinamento, gerando um hipervetor de consulta H_Q .
- **2. Cálculo de similaridade:** o sistema calcula a similaridade entre H_Q e os protótipos armazenados (P_{Real} e P_{Fake}) usando a similaridade do cosseno:

$$\text{sim}(H_Q, P_c) = (H_Q \cdot P_c) / (\|H_Q\| \cdot \|P_c\|)$$

- **3. Decisão (argmax):** a classe predita é aquela cujo protótipo apresenta a maior similaridade com o vetor da consulta:

$$\text{classe}(Q) = \text{argmax}_{c \in \{\text{Real}, \text{Fake}\}} \text{sim}(H_Q, P_c)$$

Este processo de inferência é extremamente leve, envolvendo apenas operações de produto escalar, o que viabiliza a classificação de milhares de notícias por segundo, atendendo aos requisitos de escalabilidade testados no conjunto de dados FEVER.

4.5.4 Ficha Técnica e Hiperparâmetros

A Tabela 1 consolida os hiperparâmetros utilizados em todos os classificadores. As configurações de divisão dos dados ($\text{test_size} = 30\%$, estratificação por classe, $\text{random_state} = 42$) foram adotadas como protocolo comum a todos os modelos; a execução independente

do particionamento em cada *pipeline* de codificação pode, contudo, produzir variações residuais na composição exata das partições (ver nota no Apêndice A). Os modelos de Aprendizado de Máquina clássico foram executados com os parâmetros padrão da biblioteca *scikit-learn*. Os *Transformers* (BERT e RoBERTa) foram ajustados com 2 épocas e *batch size* 16, configuração amplamente utilizada na literatura para classificação textual binária. Cabe ressaltar que a fixação de uma única semente aleatória (*random_state* = 42) prioriza a reprodutibilidade ponto a ponto dos experimentos, mas representa uma limitação metodológica: as diferenças observadas entre modelos próximos — como, por exemplo, OnlineHD (94,07%) e DistHD (93,95%) no ISOT em $D = 1.000$ — não são acompanhadas de desvio-padrão nem de teste de significância estatística. Para trabalhos futuros, recomenda-se a execução com múltiplas sementes (mínimo de cinco) e a aplicação de testes pareados (Wilcoxon ou *paired t-test*) sobre as variantes principais, de modo a fundamentar estatisticamente as comparações entre modelos com desempenhos numericamente próximos. Por decisão de escopo, a repetição com múltiplas sementes não foi realizada nesta dissertação e é registrada como trabalho futuro (Seção 6.5); a limitação correspondente é discutida na Seção 5.8.

Tabela 1 – Ficha Técnica e Hiperparâmetros dos Modelos Avaliados

Modelo	Categoria	Hiperparâmetros Específicos
<i>Vanilla</i> HDC	HDC	$D \in \{1.000; 3.000\}$; <i>bundling</i> direto sem correção de erro; epochs = 2
AdaptHD	HDC	$D \in \{1.000; 3.000\}$; aprendizado iterativo com correção de erro; epochs = 2
OnlineHD	HDC	$D \in \{1.000; 3.000\}$; epochs = 2; taxa de atualização adaptativa por similaridade
NeuralHD	HDC	$D \in \{1.000; 3.000\}$; regeneração dinâmica de dimensões; epochs = 2
DistHD	HDC	$D \in \{1.000; 3.000\}$; codificação dinâmica com decisão top-2; epochs = 2
<i>Logistic Regression</i>	Aprendizado de Máquina	$C = 1.0$; solver = lbfgs; max_iter = 100; penalty = l2
SVM	Aprendizado de Máquina	$C = 1.0$; kernel = rbf; gamma = scale; max_iter = -1
<i>Random Forest</i>	Aprendizado de Máquina	n_estimators = 100; criterion = gini; max_depth = None
Árvore de Decisão	Aprendizado de Máquina	criterion = gini; max_depth = None; min_samples_split = 2

Modelo	Categoria	Hiperparâmetros Específicos
KNN	Aprendizado de Máquina	n_neighbors = 5; weights = uniform; metric = minkowski
BERT	<i>Transformer</i>	bert-base-uncased; epochs = 2; batch = 16; lr = 2e-5; max_length = 512
RoBERTa	<i>Transformer</i>	roberta-base; epochs = 2; batch = 16; lr = 2e-5; max_length = 512

Fonte: Elaborado pelo autor (2026).

Parâmetros comuns a todos: test_size = 0,30; stratify = y; random_state = 42.

4.6 Métricas de Avaliação

As métricas de avaliação derivam da Matriz de Confusão, que quantifica:

- **Verdadeiros Positivos (TP):** *fake news* classificadas corretamente como *fake*.
- **Verdadeiros Negativos (TN):** notícias reais classificadas corretamente como reais.
- **Falsos Positivos (FP):** notícias reais classificadas incorretamente como *fake* (Erro Tipo I).
- **Falsos Negativos (FN):** *fake news* classificadas incorretamente como reais (Erro Tipo II).

A partir destes valores, calculam-se quatro das cinco métricas centrais adotadas nesta dissertação; a quinta — a área sob a curva ROC — é definida ao final desta seção. Em cenários desbalanceados (como o COVID-19), a Acurácia isolada torna-se pouco informativa; por isso, a análise privilegia as métricas por classe — Precisão, Revocação e *F1-Score* —, que permitem identificar em qual classe o classificador concentra seus erros. As métricas centrais são definidas a seguir:

$$\text{Acurácia} = (TP + TN) / (TP + TN + FP + FN),$$

$$\text{Precisão (P)} = TP / (TP + FP),$$

$$\text{Revocação (R)} = TP / (TP + FN),$$

$$\text{F1-Score} = 2 \cdot P \cdot R / (P + R).$$

O *F1-Score* é a métrica crítica para a avaliação no cenário desbalanceado do COVID-19, onde o desbalanceamento de classes (apenas 24,15% de *fakes*) torna a Acurácia uma métrica enganosa — um modelo que classifique sempre como “Real” teria aproximadamente 75,85% de acurácia, mas *F1-Score* nulo para a classe *fake*.

Por fim, dado que na detecção de *fake news* o custo dos erros é assimétrico — um falso negativo (*fake news* não sinalizada) tende a ser mais danoso que um falso positivo —, a

análise dos resultados privilegia o *F1-Score*, com atenção particular à classe minoritária (*fake*).

Às quatro métricas escalares acima soma-se a quinta métrica adotada nesta dissertação, a área sob a curva ROC, acompanhada de um instrumento diagnóstico complementar; ambos são apresentados por modelo e por conjunto de dados no Capítulo 5. A matriz de confusão exibe a distribuição completa de acertos e erros (TP, TN, FP, FN), permitindo inspecionar visualmente em qual classe cada modelo concentra suas falhas. A curva ROC (*Receiver Operating Characteristic*) relaciona a Taxa de Verdadeiros Positivos (TPR) à Taxa de Falsos Positivos (FPR), e a área sob a curva (AUC — *Area Under the Curve*) resume, em um único valor entre 0,5 (desempenho aleatório) e 1,0 (classificação perfeita), a capacidade de discriminação do modelo de forma independente do limiar de decisão. Cabe registrar que a maior parte dos classificadores avaliados produz decisões rígidas, e não escores contínuos de probabilidade. Para esses modelos, a curva ROC reduz-se a um único ponto de operação, e o AUC reportado equivale à acurácia balanceada, isto é, à média entre sensibilidade e especificidade; as curvas apresentadas no Capítulo 5 são, portanto, ROC empíricas de ponto único, ligando a origem ao ponto de operação observado e deste ao vértice (1, 1). Apenas o BERT e o RoBERTa, que emitem probabilidades por classe, têm AUC calculado por varredura contínua do limiar de decisão, o que explica seus valores sistematicamente superiores. O AUC é especialmente informativo em cenários desbalanceados, por não ser enviesado pela classe majoritária, complementando o *F1-Score* na análise da classe minoritária (*fake*).

4.7 Estimação do Consumo Energético

Para quantificar a vantagem de eficiência computacional do HDC frente aos *Transformers*, o consumo energético foi estimado a partir do TDP nominal da GPU multiplicado pelo tempo de execução, conforme a fórmula:

$$E \text{ (Wh)} = (\text{TDP} \times t) / 3.600,$$

em que TDP é a potência nominal da GPU em watts (W) — 70 W para a NVIDIA Tesla T4 utilizada —, t é o tempo total de execução em segundos (s) e E é o consumo energético resultante em watt-hora (Wh); a divisão por 3.600 converte watt-segundo (joule) em watt-hora.

Esta abordagem constitui uma estimativa de limite superior teórico, assumindo utilização constante da GPU em sua TDP nominal — premissa que não é estritamente

verdadeira para modelos rápidos (como *Logistic Regression*) que não saturam a GPU. Ressalta-se, ademais, que os modelos de referência de Aprendizado de Máquina clássico (LR, SVM, *Random Forest*, Árvore de Decisão e KNN) executam em CPU por meio do scikit-learn, de modo que a aplicação do TDP da GPU a esses modelos deve ser lida apenas como referência homogênea de limite superior, e não como medição do dispositivo efetivamente utilizado. Reconhece-se, portanto, que a estimativa superestima o consumo dos modelos mais leves; a comparação central desta dissertação — *Transformers* versus HDC — permanece informativa, pois ambas as famílias executam na mesma GPU e tendem a saturá-la durante a maior parte do tempo de execução em operações tensoriais densas. Medições diretas via NVML (nvidia-smi) ou bibliotecas como CodeCarbon constituem aprimoramento metodológico recomendado para trabalhos futuros (Seção 6.5). A estimativa por TDP foi mantida nesta dissertação por oferecer um critério homogêneo, aplicável da mesma forma a todos os modelos, inclusive aos executados em CPU, cujo consumo real tende a não ser diretamente mensurável em ambientes virtualizados como o Google Colab. Trata-se de uma estimativa, e não de uma medição, mas a direção de seu erro é conhecida: por assumir potência máxima durante todo o tempo de execução, ela representa um limite superior para todos os modelos. Se o HDC não saturar integralmente a GPU, seu consumo estará superestimado — e a vantagem energética frente ao BERT, subestimada —, de modo que as razões reportadas no Capítulo 5 devem ser lidas como conservadoras.

Pela mesma razão, os resultados são expressos em energia (Wh), e não em emissões de CO₂. A conversão para emissões dependeria da matriz energética que abastece o *datacenter* em que cada sessão do Google Colab é executada — isto é, da proporção de fontes renováveis e fósseis na eletricidade consumida —, informação que não é disponibilizada ao usuário e que varia conforme a região de alocação. As referências à sustentabilidade e à agenda de *Green AI* nesta dissertação dizem respeito, portanto, ao consumo energético dos modelos, e não à sua pegada de carbono. Registra-se, ainda, que os tempos correspondem ao tempo de relógio (*wall-clock*) de uma única execução por configuração, em ambiente compartilhado sujeito a variações de desempenho; como as diferenças entre paradigmas são de uma a três ordens de magnitude, essas variações não alteram as conclusões.

5 EXPERIMENTOS E RESULTADOS

Neste capítulo são apresentados os experimentos conduzidos e os resultados quantitativos obtidos. Diferentemente de uma organização por conjunto de dados, a análise é estruturada como um **estudo de ablação em quatro etapas encadeadas**, no qual cada experimento fixa uma decisão de projeto antes de avançar para a seguinte: primeiro compara-se a dimensionalidade hiperdimensional (Experimento 1), depois o esquema de codificação textual (Experimento 2), em seguida compara-se a melhor configuração resultante frente aos *Transformers* (Experimento 3) e, por fim, quantifica-se o custo computacional dos modelos representativos de cada paradigma (Experimento 4). Em cada experimento, os resultados são reportados em uma tabela por conjunto de dados (ISOT, COVID-19 e FEVER), totalizando o processamento de mais de 160 mil instâncias.

A discussão é estruturada em torno da hipótese central da dissertação: a Computação Hiperdimensional oferece um *trade-off favorável entre acurácia preditiva e eficiência energética*, e não — como poderia ser tentador afirmar — uma alternativa superior em precisão absoluta aos *Transformers*. Os dados empíricos confirmam esta hipótese reformulada e, ao mesmo tempo, delimitam o regime de aplicabilidade do HDC.

5.1 Delineamento dos Experimentos

Seguindo a metodologia do Capítulo 4, cada modelo de classificação (Seção 4.5) foi treinado e avaliado sobre cada corpus (Seção 4.2), com as representações textuais da Seção 4.4, sob particionamento *hold-out* 70/30 estratificado, no ambiente computacional descrito na Seção 4.1. Em vez de tratar cada conjunto de dados como um experimento isolado, a combinação exaustiva de dimensões, codificações e modelos é explorada de forma incremental, em quatro experimentos que reduzem progressivamente o espaço de configurações:

- **Experimento 1 — Seleção de Dimensionalidade (Seção 5.3):** avalia os cinco modelos HDC nas duas dimensões candidatas ($D = 1.000$ e $D = 3.000$), sob todas as codificações, para determinar a dimensão que oferece o melhor compromisso de eficácia. Esta decisão condiciona todos os experimentos subsequentes.
- **Experimento 2 — Seleção do Esquema de Codificação (Seção 5.4):** fixada a dimensão vencedora, compara os modelos HDC com os classificadores clássicos de

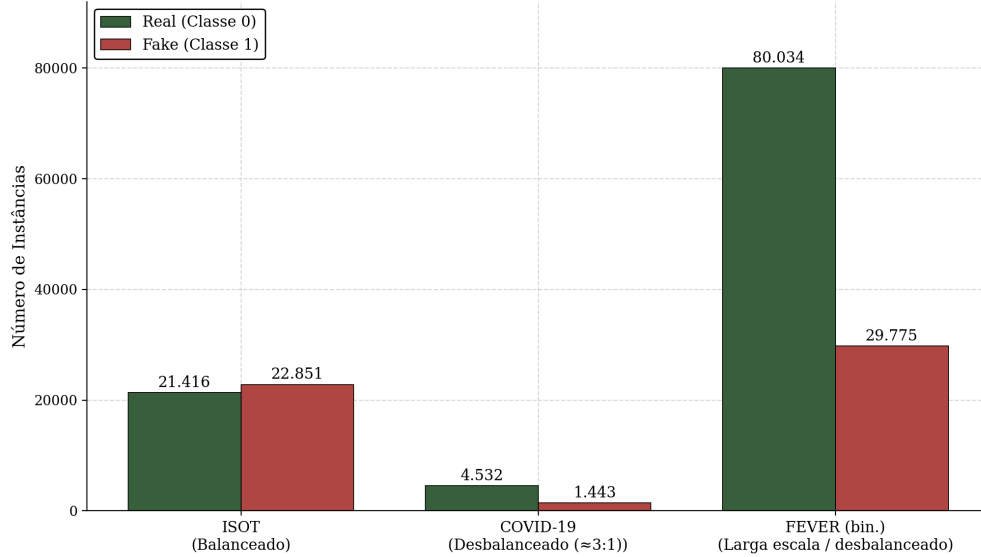
aprendizado de máquina sob cada codificação (TF-IDF, BoW e Word2Vec), identificando a representação textual mais adequada.

- **Experimento 3 — Comparação Final com *Transformers* (Seção 5.5):** com a melhor codificação selecionada, confronta todos os modelos HDC e clássicos com os *Transformers* BERT e RoBERTa, delimitando o teto de eficácia dos modelos baseados em atenção.
- **Experimento 4 — Análise de Eficiência (Seção 5.6):** para o modelo representativo de cada paradigma (um HDC, um clássico e um *Transformer*), quantifica o tempo de execução e o consumo energético estimado, consolidando o *trade-off* central da dissertação.

Os três conjuntos de dados diferem propositalmente em volume e balanceamento — ISOT (44.267 instâncias, balanceado), COVID-19 (5.975, desbalanceado $\approx 3:1$) e FEVER binarizado (109.809, larga escala) —, o que permite observar cada decisão de projeto sob condições distintas de dificuldade. Como não foi reservado conjunto de validação (Seção 4.2.3), todas as comparações dos Experimentos 1 e 2 são reportadas no conjunto de teste, e a configuração adotada como referência nos experimentos seguintes resulta da leitura dessa grade completa — aspecto registrado como limitação na Seção 5.8.

5.2 Visão Geral e Consolidação dos Dados Processados

A validação do sistema envolveu a ingestão e classificação de três corpora distintos, totalizando $N_{\text{total}} = 160.051$ amostras processadas. A Tabela 2 sintetiza a distribuição volumétrica dos dados submetidos aos classificadores, complementada pela Figura 7, que ilustra graficamente o grau de balanceamento de cada conjunto.

Figura 7 – Distribuição de classes nos três conjuntos de dados utilizados.

Fonte: Elaborado pelo autor (2026).

Tabela 2 – Distribuição Volumétrica dos Conjuntos de Dados Utilizados

Dataset	Total (N)	Classe 0 (Real)	Classe 1 (<i>Fake</i>)	Característica Principal
ISOT	44.267	21.416	22.851	Balanceado
COVID-19	5.975	4.532	1.443	Desbalanceado (≈ 3:1)
FEVER (bin.)	109.809	80.034	29.775	Larga escala / desbalanceado
Total	160.051	105.982	54.069	—

Fonte: Elaborado pelo autor (2026).

Esta heterogeneidade nos volumes — de 5.975 (COVID-19) a 109.809 (FEVER) — permitiu avaliar a estabilidade dos algoritmos em três ordens de magnitude distintas.

5.3 Experimento 1: Seleção de Dimensionalidade

O primeiro experimento investiga o impacto da dimensionalidade D dos hipervetores sobre a eficácia dos modelos HDC. A dimensionalidade é o hiperparâmetro estrutural mais relevante do paradigma: valores maiores tendem a aumentar a quase-ortogonalidade entre hipervetores aleatórios e, portanto, a capacidade de separação, mas elevam o custo de memória e de computação. Avaliaram-se duas dimensões candidatas — $D = 1.000$ e $D = 3.000$ — para as cinco variantes HDC (*Vanilla*, *AdaptHD*, *OnlineHD*, *NeuralHD* e *DistHD*), sob as três codificações. As Tabelas 3, 4 e 5 apresentam os resultados para os conjuntos ISOT, COVID-19 e FEVER, respectivamente, destacando em negrito, em cada par de dimensões, a configuração de maior *F1-Score*.

Tabela 3 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto ISOT

Codificação	Modelo	Dim.	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	Vanilla	1.000	60,93	62,45	60,95	61,69	60,90
		3.000	68,62	70,24	68,02	69,11	68,54
	AdaptHD	1.000	54,17	56,22	50,66	53,30	54,48
		3.000	59,23	59,95	63,27	61,57	59,13
	OnlineHD	1.000	89,35	89,69	89,69	89,69	89,17
		3.000	91,11	91,57	91,16	91,37	90,99
	NeuralHD	1.000	90,76	93,13	88,64	90,83	90,77
		3.000	91,51	94,16	89,07	91,54	91,52
	DistHD	1.000	90,65	90,80	91,12	90,96	90,43
		3.000	91,29	92,03	91,00	91,51	91,24
BoW	Vanilla	1.000	83,74	87,73	79,63	83,49	83,88
		3.000	84,33	88,65	79,87	84,03	84,48
	AdaptHD	1.000	87,68	87,82	88,40	88,11	87,66
		3.000	95,56	94,21	97,38	95,77	95,50
	OnlineHD	1.000	77,96	76,14	83,47	79,64	77,78
		3.000	85,31	84,99	86,88	85,92	85,25
	NeuralHD	1.000	77,46	78,98	76,77	77,86	77,48
		3.000	79,01	80,69	78,01	79,33	79,04
	DistHD	1.000	90,82	92,20	89,83	91,00	90,86
		3.000	94,43	95,77	93,33	94,54	94,47
Word2Vec	Vanilla	1.000	78,27	78,19	80,07	79,12	78,22
		3.000	83,32	86,72	79,77	83,10	83,43
	AdaptHD	1.000	72,53	87,55	54,28	67,01	73,06
		3.000	56,75	54,32	99,85	70,36	55,50
	OnlineHD	1.000	94,07	93,52	95,05	94,28	94,04
		3.000	94,53	94,62	94,74	94,68	94,52
	NeuralHD	1.000	87,37	89,08	85,98	87,50	87,41
		3.000	88,53	90,55	86,75	88,61	88,58
	DistHD	1.000	93,95	95,92	92,15	94,00	94,00
		3.000	94,53	94,70	94,67	94,68	94,53

Fonte: Elaborado pelo autor (2026).

Em cada par de dimensões (mesmo modelo e codificação), o maior F1-Score aparece em negrito.

Tabela 4 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto COVID-19

Codificação	Modelo	Dim.	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	Vanilla	1.000	77,58	52,13	69,01	59,39	70,88
		3.000	83,55	63,51	72,30	67,62	76,40
	AdaptHD	1.000	86,28	73,32	66,43	69,70	52,86
		3.000	84,27	68,37	62,91	65,53	60,28
	OnlineHD	1.000	86,28	69,23	76,06	72,48	66,68
		3.000	86,39	68,96	77,70	73,07	66,40
	NeuralHD	1.000	85,95	67,83	77,70	72,43	77,39
		3.000	86,50	69,33	77,46	73,17	76,64
	DistHD	1.000	86,61	69,62	77,46	73,33	69,09
		3.000	85,50	67,97	73,71	70,72	65,51
BoW	Vanilla	1.000	76,19	50,48	73,44	59,83	75,25
		3.000	75,63	49,68	71,13	58,50	74,10
	AdaptHD	1.000	53,59	32,29	84,06	46,66	63,98
		3.000	60,79	36,22	81,98	50,24	68,01
	OnlineHD	1.000	82,82	63,38	68,36	65,78	77,89
		3.000	83,99	66,01	69,52	67,72	79,06
	NeuralHD	1.000	74,85	48,71	78,52	60,12	76,10
		3.000	75,91	50,07	78,75	61,22	76,88
	DistHD	1.000	78,75	54,22	77,14	63,68	78,20
		3.000	81,48	60,00	69,98	64,61	77,56
Word2Vec	Vanilla	1.000	68,54	41,83	71,43	52,76	69,52
		3.000	71,45	45,05	73,24	55,79	72,05
	AdaptHD	1.000	77,41	86,00	9,75	17,52	54,62
		3.000	81,65	67,61	48,75	56,65	70,56
	OnlineHD	1.000	84,11	71,67	58,50	64,42	75,48
		3.000	85,78	72,68	67,57	70,04	79,65

Codificação	Modelo	Dim.	Acur.	Prec.	Rec.	F1	AUC
	NeuralHD	1.000	79,20	55,26	80,95	65,69	79,79
		3.000	80,26	56,96	80,73	66,79	80,42
	DistHD	1.000	84,72	75,08	56,69	64,60	75,28
		3.000	84,33	69,61	64,40	66,90	77,61

Fonte: Elaborado pelo autor (2026).

Em cada par de dimensões (mesmo modelo e codificação), o maior *F1-Score* aparece em negrito.

Tabela 5 – Estudo de Dimensionalidade dos Modelos HDC no Conjunto FEVER

Codificação	Modelo	Dim.	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	Vanilla	1.000	60,30	34,73	52,63	41,85	57,92
		3.000	63,01	37,02	51,72	43,15	59,46
	AdaptHD	1.000	63,54	31,27	28,65	29,91	52,58
		3.000	63,89	32,34	30,25	31,26	53,48
	OnlineHD	1.000	66,30	38,39	40,00	39,18	58,06
		3.000	66,90	39,25	40,10	39,67	58,55
	NeuralHD	1.000	69,52	43,66	42,40	43,02	60,99
		3.000	70,01	44,45	42,05	43,22	61,21
	DistHD	1.000	66,45	38,84	41,08	39,93	58,49
		3.000	66,64	39,07	40,96	39,99	58,61
BoW	Vanilla	1.000	57,65	33,83	58,75	42,93	58,00
		3.000	63,91	39,13	59,54	47,22	62,54
	AdaptHD	1.000	72,94	57,34	0,91	1,80	50,33
		3.000	53,33	33,71	74,61	46,44	60,01
	OnlineHD	1.000	77,35	61,96	42,65	50,52	66,46
		3.000	77,95	64,70	41,12	50,28	66,39
	NeuralHD	1.000	73,38	51,06	44,51	47,56	64,32
		3.000	75,57	56,54	42,89	48,78	65,31
	DistHD	1.000	74,42	53,35	45,20	48,94	65,25
		3.000	77,12	61,22	42,57	50,22	66,27
Word2Vec	Vanilla	1.000	56,55	33,10	58,72	42,33	57,23
		3.000	56,90	33,84	61,44	43,64	58,32
	AdaptHD	1.000	40,04	29,41	86,24	43,86	54,53
		3.000	60,59	36,20	59,11	44,90	60,13
	OnlineHD	1.000	66,48	40,72	51,33	45,41	61,73
		3.000	66,12	40,10	50,08	44,54	61,09
	NeuralHD	1.000	70,37	46,53	60,85	52,73	67,39
		3.000	71,68	48,30	60,27	53,62	68,11
	DistHD	1.000	69,20	43,56	45,24	44,38	61,69
		3.000	67,32	40,27	42,03	41,13	59,39

Fonte: Elaborado pelo autor (2026).

Em cada par de dimensões (mesmo modelo e codificação), o maior *F1-Score* aparece em negrito.

Os resultados favorecem consistentemente a maior dimensionalidade. Em comparações par a par (mesmo modelo, codificação e conjunto de dados), **D = 3.000 supera D = 1.000 em 39 dos 45 pares avaliados em *F1-Score***. Em média, D = 3.000 alcança F1 = 64,73% contra 61,14% de D = 1.000, e acurácia de 77,03% contra 75,49%. Os seis casos em que D = 1.000 se mostrou superior concentram-se nos conjuntos desbalanceados e envolvem diferenças estreitas de *F1-Score* — no máximo 4,17 pontos percentuais, e abaixo de um ponto em dois deles —, não constituindo evidência de vantagem intrínseca da dimensão menor. A ligeira elevação de custo computacional de D = 3.000 é, portanto, justificada pelo ganho sistemático de eficácia, e **esta dimensão é adotada como padrão nos experimentos subsequentes**. Cabe registrar que essa comparação foi realizada no conjunto de teste, sem conjunto de validação separado (Seção 4.2.3); a consistência da vantagem — 39 de 45 pares

— torna improvável que a escolha se alterasse sob outro protocolo, mas a limitação é discutida na Seção 5.8.

5.4 Experimento 2: Seleção do Esquema de Codificação

Fixada a dimensionalidade em $D = 3.000$, o segundo experimento compara os modelos HDC com os classificadores clássicos de aprendizado de máquina (*Logistic Regression*, *Random Forest*, SVM, Árvore de Decisão e KNN) sob cada uma das três codificações textuais. O objetivo é identificar a representação que melhor conduz o sinal linguístico para os modelos HDC: o critério de seleção considera exclusivamente o melhor *F1-Score* entre as variantes HDC, e os classificadores clássicos são incluídos nas mesmas tabelas apenas como referência, para situar o desempenho do HDC sob cada codificação. As Tabelas 6, 7 e 8 reportam os resultados para os conjuntos ISOT, COVID-19 e FEVER, respectivamente, com o melhor *F1-Score* de cada codificação em negrito.

Tabela 6 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto ISOT

Codificação	Modelo	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	<i>Vanilla</i> (HDC)	68,62	70,24	68,02	69,11	68,54
	AdaptHD (HDC)	59,23	59,95	63,27	61,57	59,13
	OnlineHD (HDC)	91,11	91,57	91,16	91,37	90,99
	NeuralHD (HDC)	91,51	94,16	89,07	91,54	91,52
	DistHD (HDC)	91,29	92,03	91,00	91,51	91,24
	<i>Logistic Regression</i>	93,72	94,66	93,27	93,96	93,32
	<i>Random Forest</i>	96,33	96,41	96,59	96,50	96,24
	SVM	94,96	96,33	93,96	95,13	94,80
	Árvore de Decisão	97,87	98,01	97,85	97,94	97,80
	KNN	81,14	80,37	84,68	82,47	80,69
BoW	<i>Vanilla</i> (HDC)	84,33	88,65	79,87	84,03	84,48
	AdaptHD (HDC)	95,56	94,21	97,38	95,77	95,50
	OnlineHD (HDC)	85,31	84,99	86,88	85,92	85,25
	NeuralHD (HDC)	79,01	80,69	78,01	79,33	79,04
	DistHD (HDC)	94,43	95,77	93,33	94,54	94,47
	<i>Logistic Regression</i>	95,98	97,32	94,91	96,09	96,01
	<i>Random Forest</i>	96,07	96,02	96,60	96,28	96,00
	SVM	80,09	82,54	78,63	80,54	79,87
	Árvore de Decisão	97,88	98,20	97,76	97,97	97,89
	KNN	76,18	80,42	72,09	76,03	76,44
Word2Vec	<i>Vanilla</i> (HDC)	83,32	86,72	79,77	83,10	83,43
	AdaptHD (HDC)	56,75	54,32	99,85	70,36	55,50
	OnlineHD (HDC)	94,53	94,62	94,74	94,68	94,52
	NeuralHD (HDC)	88,53	90,55	86,75	88,61	88,58
	DistHD (HDC)	94,53	94,70	94,67	94,68	94,53
	<i>Logistic Regression</i>	94,51	94,66	94,70	94,68	94,50
	<i>Random Forest</i>	94,88	94,91	95,19	95,05	94,87
	SVM	95,52	95,71	95,62	95,66	95,52
	Árvore de Decisão	93,91	93,77	94,55	94,15	93,89
	KNN	93,70	94,01	93,84	93,91	93,70

Fonte: Elaborado pelo autor (2026).

Modelos HDC avaliados em $D = 3.000$. O melhor *F1-Score* de cada codificação aparece em negrito.

Tabela 7 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto COVID-19

Codificação	Modelo	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	<i>Vanilla</i> (HDC)	83,55	63,51	72,30	67,62	76,40
	AdaptHD (HDC)	84,27	68,37	62,91	65,53	60,28
	OnlineHD (HDC)	86,39	68,96	77,70	73,07	66,40
	NeuralHD (HDC)	86,50	69,33	77,46	73,17	76,64
	DistHD (HDC)	85,50	67,97	73,71	70,72	65,51
	<i>Logistic Regression</i>	78,47	90,00	10,56	18,91	55,09
	<i>Random Forest</i>	81,37	89,66	24,41	38,38	61,76
	SVM	79,09	91,80	13,15	23,00	56,42
	Árvore de Decisão	82,49	65,38	55,87	60,25	73,20
	KNN	83,21	72,56	47,18	57,18	70,81
BoW	<i>Vanilla</i> (HDC)	75,63	49,68	71,13	58,50	74,10
	AdaptHD (HDC)	60,79	36,22	81,98	50,24	68,01
	OnlineHD (HDC)	83,99	66,01	69,52	67,72	79,06
	NeuralHD (HDC)	75,91	50,07	78,75	61,22	76,88
	DistHD (HDC)	81,48	60,00	69,98	64,61	77,56
	<i>Logistic Regression</i>	86,01	83,03	51,77	63,62	74,24
	<i>Random Forest</i>	80,93	86,21	23,47	36,90	61,12
	SVM	85,44	84,80	47,29	60,59	72,33
	Árvore de Decisão	81,82	63,09	56,57	59,65	72,97
	KNN	77,30	75,68	6,57	12,10	53,09
Word2Vec	<i>Vanilla</i> (HDC)	71,45	45,05	73,24	55,79	72,05
	AdaptHD (HDC)	81,65	67,61	48,75	56,65	70,56
	OnlineHD (HDC)	85,78	72,68	67,57	70,04	79,65
	NeuralHD (HDC)	80,26	56,96	80,73	66,79	80,42
	DistHD (HDC)	84,33	69,61	64,40	66,90	77,61
	<i>Logistic Regression</i>	86,17	83,78	55,36	66,67	75,89
	<i>Random Forest</i>	85,28	87,70	47,77	61,85	72,73
	SVM	87,06	88,30	55,58	68,22	76,55
	Árvore de Decisão	84,27	75,62	54,69	63,47	74,42
	KNN	81,48	63,24	61,83	62,53	74,99

Fonte: Elaborado pelo autor (2026).

Modelos HDC avaliados em $D = 3.000$. O melhor F1-Score de cada codificação aparece em negrito.

Tabela 8 – Comparação HDC × Aprendizado de Máquina Clássico por Codificação no Conjunto FEVER

Codificação	Modelo	Acur.	Prec.	Rec.	F1	AUC
TF-IDF	<i>Vanilla</i> (HDC)	63,01	37,02	51,72	43,15	59,46
	AdaptHD (HDC)	63,89	32,34	30,25	31,26	53,48
	OnlineHD (HDC)	66,90	39,25	40,10	39,67	58,55
	NeuralHD (HDC)	70,01	44,45	42,05	43,22	61,21
	DistHD (HDC)	66,64	39,07	40,96	39,99	58,61
	<i>Logistic Regression</i>	73,14	91,94	2,52	4,90	51,21
	<i>Random Forest</i>	77,38	82,78	22,35	35,20	60,24
	SVM	73,14	87,32	2,66	5,17	51,25
	Árvore de Decisão	73,61	52,71	38,70	44,63	62,67
	KNN	72,36	49,44	24,87	33,09	57,56
BoW	<i>Vanilla</i> (HDC)	63,91	39,13	59,54	47,22	62,54
	AdaptHD (HDC)	53,33	33,71	74,61	46,44	60,01
	OnlineHD (HDC)	77,95	64,70	41,12	50,28	66,39
	NeuralHD (HDC)	75,57	56,54	42,89	48,78	65,31
	DistHD (HDC)	77,12	61,22	42,57	50,22	66,27
	<i>Logistic Regression</i>	76,81	71,74	26,08	37,91	61,04
	<i>Random Forest</i>	77,82	83,49	24,08	37,37	61,10
	SVM	75,62	89,74	12,74	22,32	56,05
	Árvore de Decisão	75,18	57,29	38,06	45,73	63,61
	KNN	71,07	41,81	13,36	20,25	53,08
Word2Vec	<i>Vanilla</i> (HDC)	56,90	33,84	61,44	43,64	58,32
	AdaptHD (HDC)	60,59	36,20	59,11	44,90	60,13
	OnlineHD (HDC)	66,12	40,10	50,08	44,54	61,09
	NeuralHD (HDC)	71,68	48,30	60,27	53,62	68,11

Codificação	Modelo	Acur.	Prec.	Rec.	F1	AUC
	DistHD (HDC)	67,32	40,27	42,03	41,13	59,39
	<i>Logistic Regression</i>	79,28	74,81	36,53	49,09	65,95
	<i>Random Forest</i>	77,40	73,89	26,86	39,40	61,64
	SVM	79,06	79,48	31,59	45,21	64,26
	Árvore de Decisão	76,11	61,60	33,58	43,46	62,85
	KNN	75,55	60,00	31,84	41,60	61,92

Fonte: Elaborado pelo autor (2026).

Modelos HDC avaliados em $D = 3.000$. O melhor F1-Score de cada codificação aparece em negrito.

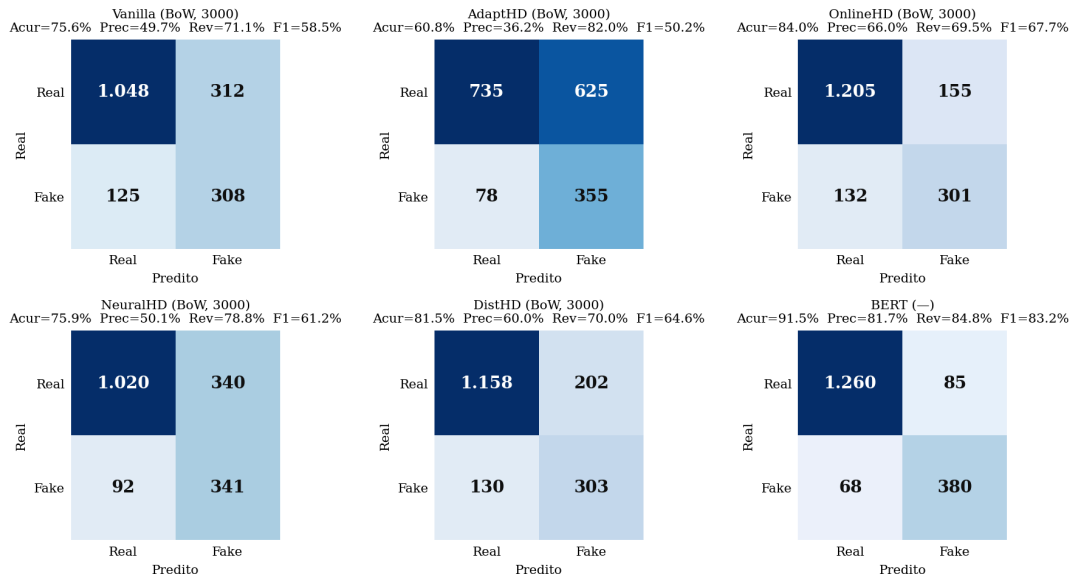
Como o critério de seleção considera apenas os modelos HDC, a análise concentra-se no melhor HDC sob cada codificação, com os clássicos como referência. Sob TF-IDF, o NeuralHD é o melhor HDC nos três conjuntos — F1 de 91,54% no ISOT, 73,17% no COVID-19 e 43,22% no FEVER —, superando todo o aprendizado de máquina clássico no COVID-19, em que a Árvore de Decisão alcança 60,25%, mas ficando 6,4 pontos percentuais atrás da Árvore de Decisão no ISOT (97,94%) e 1,4 ponto no FEVER (44,63%). Sob *Bag of Words*, o AdaptHD é o melhor HDC no ISOT (95,77%), ainda abaixo da Árvore de Decisão (97,97%), enquanto o OnlineHD é o melhor HDC — e o melhor modelo global — no COVID-19 (F1 = 67,72%) e no FEVER (F1 = 50,28%). Sob Word2Vec, o OnlineHD atinge F1 = 94,68% no ISOT, a poucos pontos do melhor clássico (SVM, 95,66%), e lidera o COVID-19 (70,04%), enquanto o NeuralHD lidera o FEVER (53,62%), o maior F1 obtido pelo HDC nesse conjunto. Nenhuma codificação é, portanto, superior em todos os cenários: o maior F1 HDC do ISOT ocorre sob BoW, o do COVID-19 sob TF-IDF e o do FEVER sob Word2Vec.

Agregando o melhor F1 HDC por conjunto de dados, o Word2Vec conduz o paradigma ao patamar médio mais alto (72,78% contra 71,26% do BoW e 69,31% do TF-IDF). Sob esse esquema, o melhor HDC supera o melhor modelo clássico nos dois conjuntos desbalanceados — COVID-19 (70,04% contra 68,22% do SVM) e FEVER (53,62% contra 49,09% da *Logistic Regression*) —, enquanto no ISOT balanceado o SVM permanece à frente (95,66% contra 94,68%). Como observado, os melhores resultados isolados do ISOT e do COVID-19 ocorrem sob outras codificações; o Word2Vec é adotado por oferecer o desempenho médio mais alto e a maior consistência entre os três cenários. **O Word2Vec é, portanto, adotado como a codificação de referência** para a comparação final. Diferentemente do que ocorre com a dimensionalidade, essa escolha repousa sobre diferenças modestas — 1,52 ponto percentual de F1 médio sobre o BoW —, obtidas no conjunto de teste; não se pode afirmar, portanto, que o Word2Vec seja superior às demais codificações de forma generalizável, e sua adoção deve ser entendida como a definição de uma configuração de referência para o Experimento 3 (Seção 5.8). As Figuras 8 a 13 detalham o comportamento

diagnóstico dos modelos em cada conjunto de dados — matrizes de confusão e curvas ROC com a classe *Fake* definida como positiva.

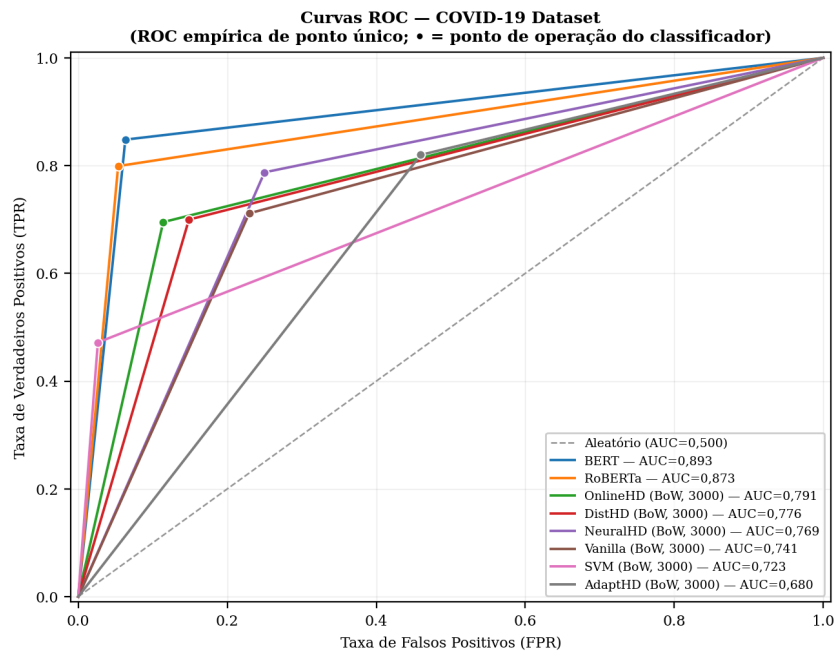
Cabe uma observação metodológica válida para todas as matrizes de confusão e curvas ROC deste capítulo (Figuras 8 a 13): para assegurar comparabilidade visual direta entre todos os modelos sob uma mesma condição, essas figuras adotam uniformemente a representação *Bag of Words* com $D = 3.000$. Por esse motivo, os valores absolutos exibidos nelas não são diretamente comparáveis aos das tabelas, que reportam a melhor configuração de cada modelo. Além disso, as figuras incluem, a título exploratório, o *baseline Vanilla* HDC e, no caso do ISOT, uma Árvore de Decisão, exibidos apenas para contextualização visual.

Matrizes de Confusão — COVID-19 Dataset (classe positiva = Fake)



Fonte: Elaborado pelo autor (2026).

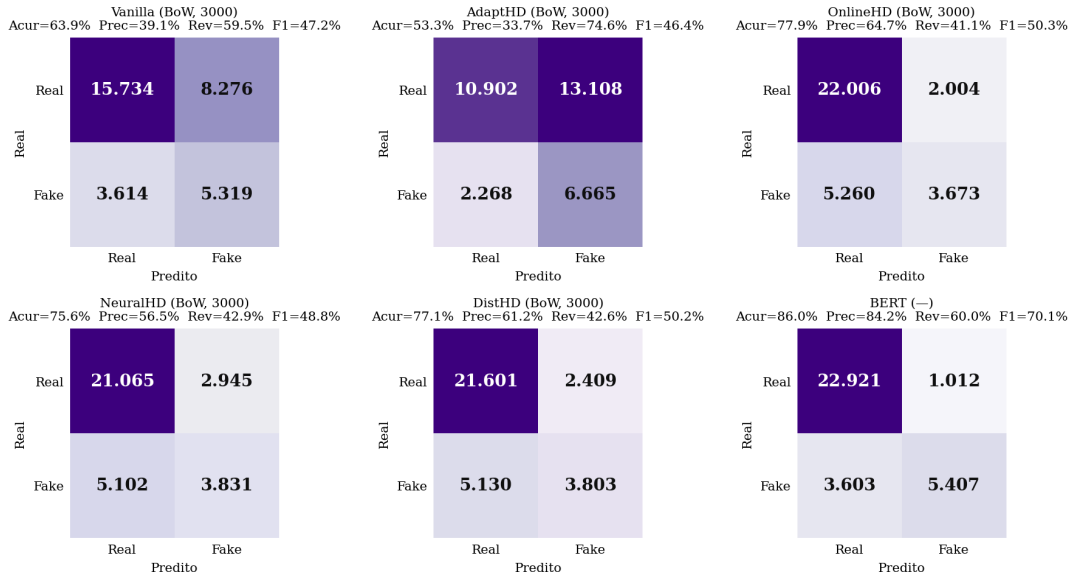
Figura 11 – Curvas ROC no conjunto de dados COVID-19 (classe positiva = Fake).



Fonte: Elaborado pelo autor (2026).

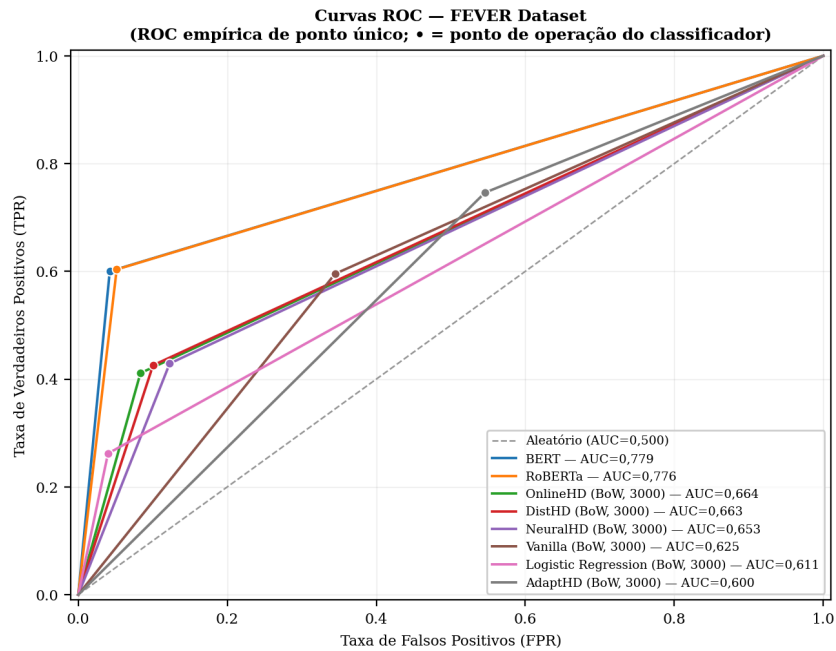
Figura 12 – Matrizes de confusão no conjunto de dados FEVER (BoW, D = 3000).

Matrizes de Confusão — FEVER Dataset (classe positiva = Fake)



Fonte: Elaborado pelo autor (2026).

Figura 13 – Curvas ROC no conjunto de dados FEVER (classe positiva = Fake).



Fonte: Elaborado pelo autor (2026).

5.5 Experimento 3: Comparação Final com *Transformers*

Com a dimensionalidade ($D = 3.000$) e a codificação (Word2Vec) fixadas, o terceiro experimento confronta todos os modelos HDC e clássicos com os *Transformers* BERT e RoBERTa. As Tabelas 9, 10 e 11 consolidam a comparação para os conjuntos ISOT, COVID-19 e FEVER, respectivamente, com o melhor *F1-Score* de cada tabela em negrito.

Tabela 9 – Comparação Final com Transformers no Conjunto ISOT (Word2Vec)

Paradigma	Modelo	Acur.	Prec.	Rec.	F1	AUC
HDC (D = 3.000)	<i>Vanilla</i>	83,32	86,72	79,77	83,10	83,43
	AdaptHD	56,75	54,32	99,85	70,36	55,50
	OnlineHD	94,53	94,62	94,74	94,68	94,52
	NeuralHD	88,53	90,55	86,75	88,61	88,58
	DistHD	94,53	94,70	94,67	94,68	94,53
ML clássico	<i>Logistic Regression</i>	94,51	94,66	94,70	94,68	94,50
	<i>Random Forest</i>	94,88	94,91	95,19	95,05	94,87
	SVM	95,52	95,71	95,62	95,66	95,52
	Árvore de Decisão	93,91	93,77	94,55	94,15	93,89
	KNN	93,70	94,01	93,84	93,91	93,70
Transformers	BERT	99,79	99,78	99,81	99,80	99,99
	RoBERTa	99,73	99,74	99,74	99,74	99,99

Fonte: Elaborado pelo autor (2026).

Modelos HDC e clássicos avaliados com codificação Word2Vec; HDC em D = 3.000. O melhor F1-Score da tabela aparece em negrito.

Tabela 10 – Comparação Final com Transformers no Conjunto COVID-19 (Word2Vec)

Paradigma	Modelo	Acur.	Prec.	Rec.	F1	AUC
HDC (D = 3.000)	<i>Vanilla</i>	71,45	45,05	73,24	55,79	72,05
	AdaptHD	81,65	67,61	48,75	56,65	70,56
	OnlineHD	85,78	72,68	67,57	70,04	79,65
	NeuralHD	80,26	56,96	80,73	66,79	80,42
	DistHD	84,33	69,61	64,40	66,90	77,61
ML clássico	<i>Logistic Regression</i>	86,17	83,78	55,36	66,67	75,89
	<i>Random Forest</i>	85,28	87,70	47,77	61,85	72,73
	SVM	87,06	88,30	55,58	68,22	76,55
	Árvore de Decisão	84,27	75,62	54,69	63,47	74,42
	KNN	81,48	63,24	61,83	62,53	74,99
Transformers	BERT	91,47	81,72	84,82	83,24	96,30
	RoBERTa	90,96	83,26	79,91	81,55	95,28

Fonte: Elaborado pelo autor (2026).

Modelos HDC e clássicos avaliados com codificação Word2Vec; HDC em D = 3.000. O melhor F1-Score da tabela aparece em negrito.

Tabela 11 – Comparação Final com Transformers no Conjunto FEVER (Word2Vec)

Paradigma	Modelo	Acur.	Prec.	Rec.	F1	AUC
HDC (D = 3.000)	<i>Vanilla</i>	56,90	33,84	61,44	43,64	58,32
	AdaptHD	60,59	36,20	59,11	44,90	60,13
	OnlineHD	66,12	40,10	50,08	44,54	61,09
	NeuralHD	71,68	48,30	60,27	53,62	68,11
	DistHD	67,32	40,27	42,03	41,13	59,39
ML clássico	<i>Logistic Regression</i>	79,28	74,81	36,53	49,09	65,95
	<i>Random Forest</i>	77,40	73,89	26,86	39,40	61,64
	SVM	79,06	79,48	31,59	45,21	64,26
	Árvore de Decisão	76,11	61,60	33,58	43,46	62,85
	KNN	75,55	60,00	31,84	41,60	61,92
Transformers	BERT	85,99	84,23	60,01	70,09	90,03
	RoBERTa	85,41	81,50	60,36	69,35	88,89

Fonte: Elaborado pelo autor (2026).

Modelos HDC e clássicos avaliados com codificação Word2Vec; HDC em D = 3.000. O melhor F1-Score da tabela aparece em negrito.

O resultado é inequívoco quanto à eficácia: o **BERT obtém o melhor F1-Score nos três conjuntos de dados** (99,80% no ISOT, 83,24% no COVID-19 e 70,09% no FEVER), seguido de perto pelo RoBERTa. A distância para o HDC, contudo, depende fortemente do cenário. No **ISOT** balanceado, a diferença é pequena: o OnlineHD e o DistHD atingem F1 =

94,68% e o SVM clássico chega a 95,66%, a cerca de cinco pontos do teto dos *Transformers* — regime em que o HDC é competitivo. Cabe observar que o próprio teto de 99,80% do BERT no ISOT deve ser lido com cautela, à luz da ameaça à validade discutida na Seção 5.8: há indícios de que os modelos treinados nesse corpus explorem marcadores estilísticos da agência Reuters, fonte concentrada da classe “real”, o que tende a inflar o desempenho de todos os paradigmas. Por essa razão, as conclusões centrais desta dissertação não se apoiam isoladamente no ISOT: os conjuntos COVID-19 e FEVER, que não apresentam esse artefato, reproduzem o mesmo padrão — superioridade de eficácia dos *Transformers* e vantagem de eficiência do HDC em relação a eles. Nos cenários **desbalanceados**, o *gap* cresce: no COVID-19, o BERT (83,24%) supera o melhor HDC (OnlineHD, 70,04%) por aproximadamente 13 pontos; no FEVER, o BERT (70,09%) supera o melhor HDC (NeuralHD, 53,62%) por aproximadamente 16 pontos.

O baixo F1 absoluto de todos os modelos no FEVER — inclusive do BERT, abaixo de 72% — reflete a dificuldade intrínseca da tarefa em configuração *claim-only*: afirmações curtas e factuais cuja veracidade depende de conhecimento externo não contido no próprio texto. A formulação original da tarefa (THORNE et al., 2018) prevê a recuperação de sentenças de evidência da Wikipédia, e problema análogo motivou Alhindi, Petridis e Muresan (2018) a propor o LIAR-PLUS com *Justification Modeling*. Para o HDC puro, que opera sobre padrões léxico-sintáticos sem acesso a bases de conhecimento externas, este cenário é particularmente desfavorável.

Em síntese, os *Transformers* preservam a liderança em eficácia em todos os cenários, mas o HDC — em especial o OnlineHD com Word2Vec — mantém-se competitivo no cenário balanceado. Resta quantificar o preço computacional dessa eficácia, objeto do experimento seguinte.

5.6 Experimento 4: Análise de Eficiência — Tempo e Energia

O quarto experimento desloca o eixo da eficácia para a eficiência. Selecionou-se o modelo representativo de cada paradigma — o campeão HDC (OnlineHD com Word2Vec), um representante clássico de baixíssimo custo (*Logistic Regression* com Word2Vec) e o *Transformer* de melhor desempenho (BERT) — e mediu-se, para cada um, o tempo de treinamento, o tempo de inferência sobre o conjunto de teste e o consumo energético estimado de cada etapa nos três conjuntos de dados. Os tempos referem-se exclusivamente à etapa correspondente — o ajuste do modelo, no treinamento, e a classificação do conjunto de teste,

na inferência — e foram medidos sobre os dados já pré-processados (Seção 4.3), sem incluir a etapa de limpeza. A energia é estimada por $E = \text{TDP} \times t / 3600$, conforme a Seção 4.7 (GPU Tesla T4, TDP de 70 W). Por comparar os três paradigmas entre si, as Tabelas 12 (treinamento) e 13 (inferência) são apresentadas de forma consolidada, agrupada por conjunto de dados. A *Logistic Regression* é adotada como representante clássico por dispor de medições completas nos três conjuntos e por exemplificar o piso de custo dos métodos tradicionais; entre os clássicos, o SVM obteve F1 marginalmente superior no cenário balanceado (Tabela 9), sem alterar as conclusões qualitativas desta análise.

Tabela 12 – Eficiência dos Modelos Representativos de Cada Paradigma (D = 3.000, Word2Vec)

Dataset	Modelo	F1 (%)	Tempo de treino (s)	Energia de treino (Wh)
ISOT	OnlineHD + W2V (HDC)	94,68	3,72	0,073
	<i>Logistic Regression</i> + W2V (ML)	94,68	0,50	0,010
	BERT (<i>Transformer</i>)	99,80	4.357,80	84,73
COVID-19	OnlineHD + W2V (HDC)	70,04	6,58	0,13
	<i>Logistic Regression</i> + W2V (ML)	66,67	0,04	0,0008
	BERT (<i>Transformer</i>)	83,24	123,60	2,40
FEVER	OnlineHD + W2V (HDC)	44,54	119,80	2,33
	<i>Logistic Regression</i> + W2V (ML)	49,09	1,08	0,021
	BERT (<i>Transformer</i>)	70,09	2.532,40	49,24

Fonte: Elaborado pelo autor (2026).

F1-Score conforme a rodada final de avaliação (Tabelas 6 a 11); tempo e energia conforme as medições de treinamento registradas no mesmo ambiente (Seção 4.1), com energia estimada via $\text{TDP} \times \text{tempo}$ (Seção 4.7).

A tabela quantifica o *trade-off* central da dissertação. No **ISOT**, o BERT consumiu 84,73 Wh (4.357,8 s $\approx$ 1h12min) para alcançar F1 = 99,80%, enquanto o OnlineHD consumiu apenas 0,073 Wh (3,72 s) para F1 = 94,68% — uma **redução de aproximadamente 1.161×** no consumo energético ao custo de cerca de 5 pontos percentuais de F1. A *Logistic Regression* é ainda mais econômica (0,010 Wh), evidenciando que, no regime balanceado, tanto o HDC quanto os clássicos operam a uma fração ínfima do custo dos *Transformers*. Nos cenários desbalanceados, a vantagem energética do HDC persiste, ainda que em fator menor: **aproximadamente 18×** no **COVID-19** (0,13 Wh vs 2,40 Wh) e **aproximadamente 21×** no **FEVER** (2,33 Wh vs 49,24 Wh).

A comparação com a *Logistic Regression*, contudo, impõe uma qualificação importante. A vantagem energética do HDC é expressiva frente aos *Transformers*, mas não frente aos classificadores clássicos: no ISOT, a *Logistic Regression* com Word2Vec alcança o mesmo *F1-Score* do OnlineHD (94,68%) com consumo cerca de sete vezes menor (0,010 Wh contra 0,073 Wh); no FEVER, supera o OnlineHD tanto em F1 (49,09% contra 44,54%) quanto em custo (0,021 Wh contra 2,33 Wh). Apenas no COVID-19 o HDC oferece ganho de

eficácia sobre o representante clássico (70,04% contra 66,67%), ainda assim a um custo energético superior (0,13 Wh contra 0,0008 Wh). Em consumo energético e em tempo, tanto no treinamento quanto na inferência (Tabelas 12 e 13), o HDC fica em segundo lugar nos três conjuntos, atrás da *Logistic Regression* e à frente do BERT. Trata-se, portanto, de uma posição intermediária no *trade-off*: muito mais eficiente que os *Transformers* e com eficácia comparável à dos classificadores clássicos, mas sem vantagem energética sobre estes no protocolo avaliado.

O custo de inferência dos mesmos modelos, medido sobre o conjunto de teste completo de cada corpus, é apresentado na Tabela 13.

Tabela 13 – Custo de Inferência dos Modelos Representativos de Cada Paradigma (D = 3.000, Word2Vec)

Dataset	Modelo	Tempo total (s)	Por instância (ms)	Energia (Wh)
ISOT	OnlineHD + W2V (HDC)	1,212	0,091	0,024
	<i>Logistic Regression</i> + W2V (ML)	0,016	0,001	0,0003
	BERT (<i>Transformer</i>)	103,902	7,823	2,020
COVID-19	OnlineHD + W2V (HDC)	0,171	0,095	0,0033
	<i>Logistic Regression</i> + W2V (ML)	0,007	0,004	0,0001
	BERT (<i>Transformer</i>)	4,646	2,591	0,090
FEVER	OnlineHD + W2V (HDC)	3,447	0,105	0,067
	<i>Logistic Regression</i> + W2V (ML)	0,038	0,001	0,0007
	BERT (<i>Transformer</i>)	58,275	1,769	1,133

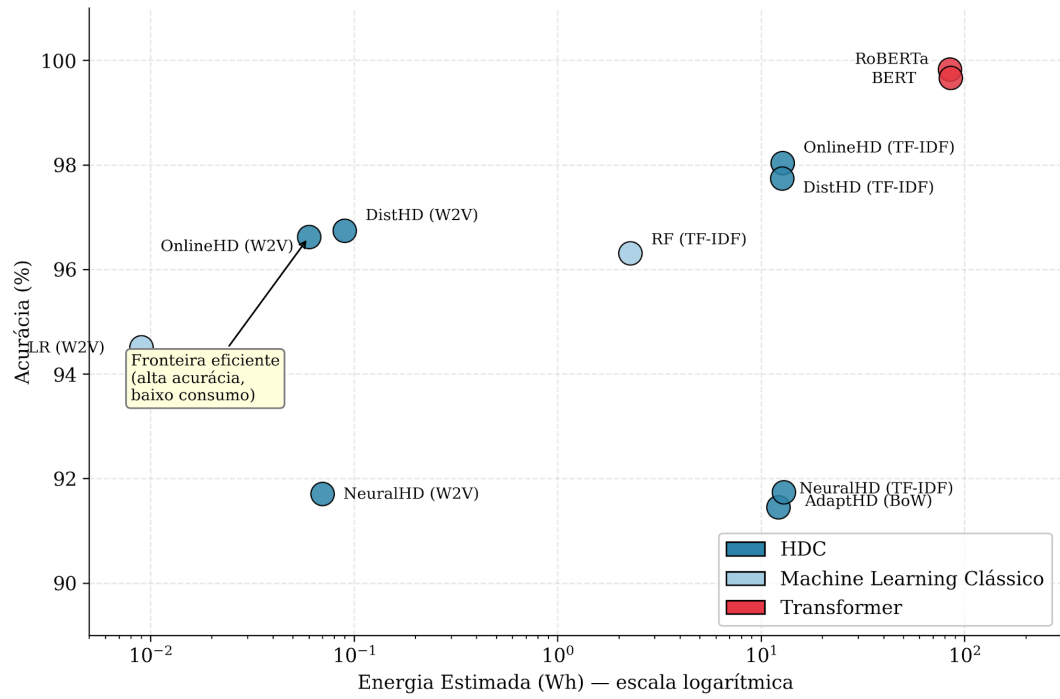
Fonte: Elaborado pelo autor (2026).

Tempo total de classificação do conjunto de teste (ISOT: 13.281; COVID-19: 1.793; FEVER: 32.943 instâncias), medido sobre os dados já pré-processados no mesmo ambiente (Seção 4.1); energia estimada via $TDP \times tempo$ (Seção 4.7).

Na inferência, a ordenação entre os paradigmas se mantém. O OnlineHD classificou os conjuntos de teste de 17 a 86 vezes mais rápido que o BERT — cerca de 0,1 ms por instância em todos os corpora, contra 1,8 a 7,8 ms do BERT —, com consumo energético proporcionalmente menor. A vantagem na inferência é, contudo, inferior à observada no treinamento no ISOT e no FEVER, pois o custo do BERT se concentra sobretudo no ajuste fino: no ISOT, a razão de tempo entre BERT e OnlineHD cai de cerca de $1.171\times$ no treinamento para cerca de $86\times$ na inferência. A *Logistic Regression*, por sua vez, foi de 23 a 91 vezes mais rápida que o OnlineHD também nessa etapa, o que confirma que a vantagem de eficiência do HDC se verifica em relação aos *Transformers*, e não aos classificadores clássicos. Em termos absolutos, os três paradigmas classificam cada instância em poucos milissegundos ou menos; a diferença torna-se relevante em cenários de grande volume de dados ou em dispositivos com restrição energética.

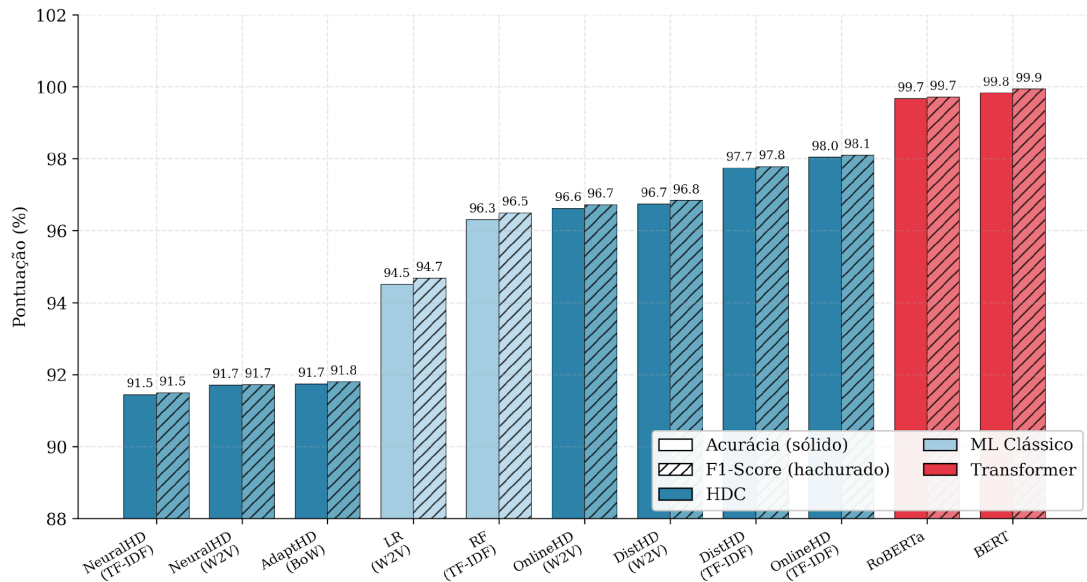
As Figuras 14 e 15 visualizam graficamente esse compromisso no conjunto ISOT — consumo energético versus acurácia e os indicadores de Acurácia e *F1-Score* por modelo.

Figura 14 – Trade-off energia × acurácia no conjunto de dados ISOT.



Fonte: Elaborado pelo autor (2026).

Figura 15 – Acurácia e *F1-Score* por modelo no conjunto de dados ISOT.



Fonte: Elaborado pelo autor (2026).

Esses números sustentam três achados centrais. **Primeiro**, os *Transformers* preservam a maior eficácia absoluta em todos os cenários. **Segundo**, os modelos HDC oferecem desempenho competitivo no cenário balanceado, a poucos pontos do teto. **Terceiro**, e decisivo

para a hipótese desta dissertação, a vantagem do HDC em relação aos *Transformers* é uma eficiência energética de ordens de magnitude — de aproximadamente $18\times$ a mais de $1.000\times$ frente ao BERT no treinamento, e de $17\times$ a $86\times$ na inferência, conforme o conjunto de dados —, que é precisamente o *trade-off* favorável previsto. Essa vantagem, contudo, não se estende aos classificadores clássicos, que operam com custo igual ou inferior e eficácia comparável.

5.7 Síntese Comparativa: O *Trade-off* Acurácia $\times$ Eficiência

A análise transversal dos quatro experimentos permite estabelecer uma síntese empírica do *trade-off* central.

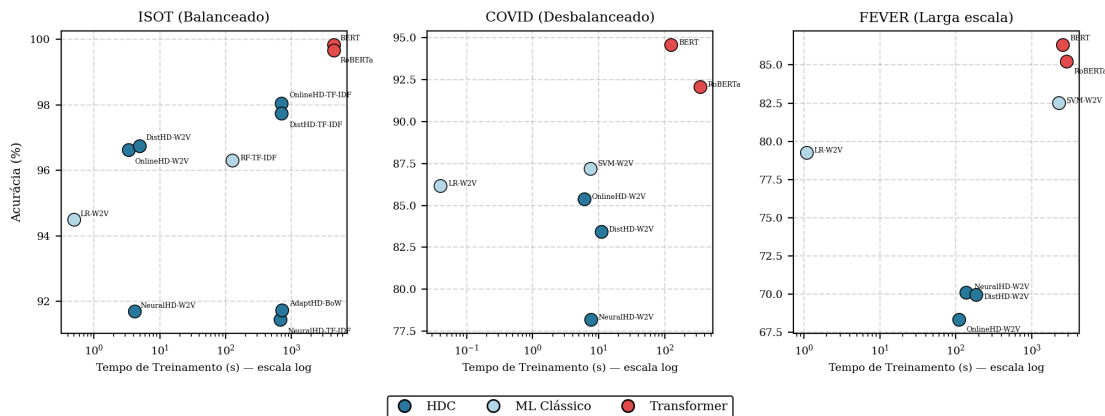
5.7.1 Complexidade de Tempo e Treinamento

Sendo N o número de amostras e L o comprimento médio do texto:

- ***Transformers (Self-Attention)***: complexidade quadrática em relação ao comprimento da sequência, $O(N \cdot L^2)$.
- **HDC**: complexidade linear em relação ao número de amostras e comprimento, $O(N \cdot L)$.

Empiricamente, esta diferença teórica traduziu-se em fatores observados de $18\times$ a mais de $1.000\times$ de redução em tempo e energia em relação ao BERT no treinamento, dependendo da combinação modelo/representação/conjunto de dados. A Figura 16 sintetiza o *trade-off* entre tempo de treinamento e acurácia para todos os modelos e conjuntos de dados.

Figura 16 – *Trade-off* tempo de treinamento $\times$ acurácia em todos os conjuntos de dados.



Fonte: Elaborado pelo autor (2026).

5.7.2 Robustez ao Desbalanceamento

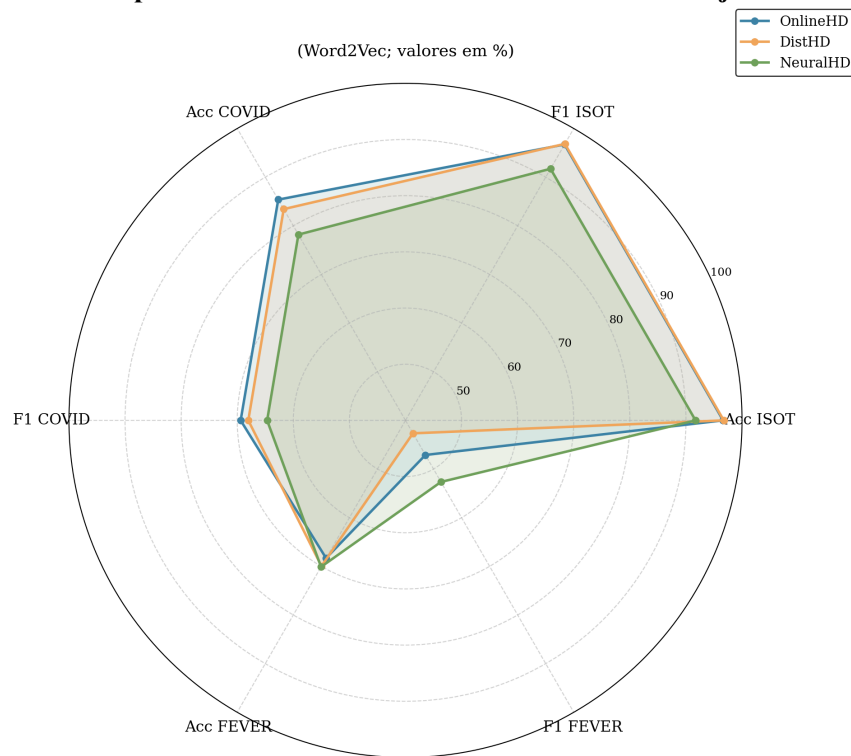
Nos cenários desbalanceados, as melhores variantes HDC superaram os modelos de referência clássicos por margens modestas: no COVID-19, o OnlineHD alcançou *F1-Score* de

70,04% contra 68,22% do SVM; no FEVER, o NeuralHD atingiu 53,62% contra 49,09% da *Logistic Regression*. A vantagem existe, mas é estreita no primeiro caso e da ordem de quatro pontos percentuais no segundo — muito aquém dos 83,24% e 70,09% obtidos pelo BERT nos mesmos conjuntos. Os dados, portanto, não sustentam nem a tese forte de robustez intrínseca ao desbalanceamento frequentemente atribuída ao paradigma na literatura, nem sua negação: o HDC opera na mesma faixa dos classificadores clássicos, com ligeira vantagem, e a distância para o estado da arte permanece substancial. O diferencial do paradigma neste regime continua sendo a eficiência operacional, não a eficácia preditiva.

5.7.3 Mapa de Aplicabilidade

Os resultados permitem mapear três regimes de uso recomendado para o HDC, sintetizados no comparativo radar das variantes HDC apresentado na Figura 17:

Figura 17 – Comparativo radar das variantes HDC nos três conjuntos de dados.



Fonte: Elaborado pelo autor (2026).

- **Regime ideal — conjuntos balanceados com textos longos:** HDC oferece desempenho próximo aos *Transformers* com redução energética superior a 1.000×. Recomenda-se OnlineHD com Word2Vec.
- **Regime intermediário — conjuntos desbalanceados ou de larga escala:** HDC mantém vantagem de eficiência ($\approx 18\text{--}21\times$) com perda significativa de F1. Recomenda-se como filtro de primeira linha em arquiteturas híbridas.

- **Regime desfavorável — tarefas que exigem raciocínio sobre evidências externas:** HDC puro encontra teto de eficácia. Recomenda-se complementar com extração de evidências explícitas (LIAR-PLUS) ou *Transformers* como segundo estágio.

5.8 Limitações Observadas

Apesar da clareza dos achados, onze limitações merecem registro explícito:

- **Ausência de modelagem de evidências:** o HDC classifica com base na “forma” estatística do texto, enquanto modelos semânticos profundos podem inferir contradições lógicas. Para tarefas que exigem raciocínio sobre evidências externas — como no LIAR-PLUS com *Justification Modeling* — o HDC puro tem limitações de eficácia, o que se reflete diretamente nos resultados do FEVER.
- **Estimativa energética por TDP:** a fórmula $E = \text{TDP} \times t / 3600$ superestima o consumo dos modelos rápidos (LR, SVM), que não saturam a GPU; o TDP da GPU foi aplicado como referência homogênea de limite superior, sendo a estimativa conservadora a favor dos *Transformers*. Medições diretas via NVML/CodeCarbon seriam mais precisas. As emissões de CO₂ não foram estimadas, por ser desconhecida a matriz energética do *datacenter* do Colab (Seção 4.7).
- **Hardware único:** os experimentos foram conduzidos exclusivamente em GPU Tesla T4. A promessa de “*Edge AI*” do HDC — execução em CPU, FPGA ou *chip* neuromórfico — não foi validada empiricamente, constituindo direção de trabalho futuro.
- **Teto de dimensionalidade:** todos os modelos HDC foram avaliados exclusivamente em $D = 1.000$ e $D = 3.000$, por limitação computacional do ambiente experimental (Seção 4.4.2). Como o Experimento 1 evidenciou ganho sistemático ao passar de $D = 1.000$ para $D = 3.000$, não se pode descartar que dimensionalidades superiores — como o $D = 10.000$ usual na literatura — reduzissem ainda mais o *gap* de eficácia em relação aos *Transformers*, ao custo de maior consumo de memória e energia. A curva de saturação da dimensionalidade não foi, portanto, caracterizada neste trabalho.
- **Orçamento de treinamento restrito:** o número de épocas (2 para todos os modelos) e o *batch size* dos *Transformers* (16) foram determinados pelas restrições de memória e de tempo de sessão do ambiente experimental, e não por busca de hiperparâmetros. Não se pode descartar que treinamentos mais longos elevassem a eficácia de ambos os

paradigmas; como o orçamento foi idêntico para HDC e *Transformers*, contudo, a comparação de eficiência permanece internamente válida.

- **Ameaça à validade externa no ISOT:** o alto desempenho no ISOT (acurácia > 96% mesmo em modelos clássicos) deve ser interpretado com cautela. Há indícios de que classificadores treinados no ISOT capturem sobretudo marcadores estilísticos da Reuters — fonte concentrada da classe “real” — e não propriamente o sinal de veracidade, o que constitui uma ameaça à validade externa. A generalização para outras fontes não foi avaliada e fica como trabalho futuro.
- **Heterogeneidade dos codificadores entre variantes HDC:** conforme registrado na Seção 4.4.2, as cinco variantes avaliadas não compartilham o mesmo codificador — *Vanilla* HDC e *AdaptHD* empregam codificação baseada em registros, *OnlineHD* e *NeuralHD* utilizam projeção aleatória não linear e *DistHD*, projeção aleatória linear. As diferenças de desempenho observadas entre elas refletem, portanto, a combinação do mecanismo de atualização com o codificador, não sendo possível isolar a contribuição de cada fator. A comparação entre paradigmas — HDC, Aprendizado de Máquina clássico e *Transformers* — não é afetada, pois cada variante é avaliada como sistema completo sobre as mesmas representações de entrada.
- **Execução com semente única:** cada configuração foi avaliada em uma única partição (random_state = 42), sem repetição com múltiplas sementes nem testes de significância estatística. Diferenças pequenas entre modelos de desempenho próximo — como *OnlineHD* (94,07%) e *DistHD* (93,95%) no ISOT em D = 1.000 — não devem, portanto, ser interpretadas como superioridade estatisticamente estabelecida. As conclusões centrais da dissertação apoiam-se em diferenças de maior magnitude, como a vantagem de eficácia dos *Transformers* e as diferenças de uma a três ordens de grandeza em eficiência.
- **Stemming antes do Word2Vec:** como o *stemming* foi aplicado antes da consulta aos vetores pré-treinados do GoogleNews, cujo vocabulário é formado por palavras completas, parte dos termos reduzidos a radicais pode não ter encontrado vetor correspondente, empobrecendo a representação dos documentos; a proporção de termos fora do vocabulário não foi medida. O efeito tende a subestimar o desempenho da codificação Word2Vec, tanto para o HDC quanto para os modelos clássicos. Como o Word2Vec foi, ainda assim, a codificação de melhor desempenho médio (Seção 5.4), sua adoção como configuração de referência não é enfraquecida por essa limitação.

- **Pré-processamento uniforme para todos os modelos:** o mesmo protocolo de limpeza, remoção de *stopwords* e *stemming* foi aplicado também aos *Transformers*, que usualmente operam sobre texto bruto (Seção 4.3). A escolha garante a igualdade de informação entre os modelos, mas tende a subestimar a eficácia dos *Transformers*; como estes lideraram todos os cenários mesmo nessa condição, a conclusão central da dissertação não é afetada.
- **Ausência de conjunto de validação:** o particionamento adotou apenas treino e teste (70/30), e as comparações dos Experimentos 1 e 2 — que definiram a dimensionalidade ($D = 3.000$) e a codificação (Word2Vec) usadas na comparação final — foram realizadas no conjunto de teste. Como o teste orientou essas escolhas, os resultados da configuração de referência podem conter viés otimista. O risco é baixo para a dimensionalidade, cuja vantagem foi consistente (39 de 45 pares), e maior para a codificação, cuja vantagem média sobre o BoW foi de apenas 1,52 ponto percentual. Como os modelos clássicos e os *Transformers* não passaram por seleção de hiperparâmetros, o eventual viés favorece apenas o HDC: isso torna conservadoras as conclusões sobre sua desvantagem de eficácia frente aos *Transformers*, mas pode ter ampliado as pequenas vantagens do HDC sobre os classificadores clássicos nos cenários desbalanceados (Seção 5.7.2), que devem ser lidas com essa ressalva.

Com esta seção, encerramos a análise dos resultados. Demonstramos empiricamente que o HDC sustenta o *trade-off* proposto — alta eficiência energética com desempenho competitivo em cenário balanceado —, mas reconhecemos os limites de sua aplicabilidade em cenários desbalanceados e que exigem raciocínio profundo.

6 CONCLUSÃO

A presente dissertação investigou a viabilidade da Computação Hiperdimensional (HDC) como paradigma algorítmico alternativo para detecção de *fake news* em cenários de *Big Data*. Motivada pela crescente insustentabilidade computacional dos modelos do Estado da Arte dominados por arquiteturas de *Deep Learning* e *Large Language Models*, esta pesquisa buscou validar empiricamente o *trade-off* entre acurácia preditiva e eficiência energética da álgebra de vetores de alta dimensão.

Ao confrontar cinco variantes de HDC com modelos de referência de Aprendizado de Máquina clássico e com *Transformers* (BERT e RoBERTa) em três conjuntos de dados de referência (ISOT, COVID-19 e FEVER), totalizando 160.051 instâncias processadas, foi possível traçar um panorama robusto e honesto sobre as capacidades e limitações desta tecnologia frente aos desafios contemporâneos da desinformação.

6.1 Revisitação da Hipótese e Síntese dos Resultados

Convém retomar a hipótese formulada na Introdução: *a Computação Hiperdimensional (HDC) constitui um paradigma alternativo capaz de oferecer um trade-off favorável entre acurácia preditiva e eficiência energética na detecção de fake news em larga escala, mantendo desempenho competitivo em cenários balanceados e reduzindo o consumo computacional em ordens de magnitude em relação a arquiteturas Transformer*. A análise dos resultados dos quatro experimentos confirma essa hipótese — com qualificações importantes: o HDC mostrou-se ferramenta computacionalmente eficiente e escalável para triagem de desinformação em larga escala, ao custo de perda de acurácia que varia de modesta (em cenários balanceados) a significativa (em cenários desbalanceados ou que exigem raciocínio sobre evidências externas). As subseções seguintes sintetizam os quatro principais achados empíricos que sustentam essa conclusão.

Antes dos achados relativos ao *trade-off*, cabe registrar os dois resultados de seleção de configuração produzidos pelos Experimentos 1 e 2, que condicionaram todo o restante da avaliação. Quanto à dimensionalidade, $D = 3.000$ superou $D = 1.000$ em 39 dos 45 pares comparados, com médias de *F1-Score* de 64,73% contra 61,14% e de acurácia de 77,03% contra 75,49%, o que justificou sua adoção como padrão. Quanto à representação textual, o Word2Vec conduziu os modelos HDC ao patamar médio mais alto (72,78% de *F1-Score* do melhor HDC por conjunto, contra 71,26% do *Bag of Words* e 69,31% do TF-IDF), superando

o melhor modelo clássico nos dois conjuntos desbalanceados — ainda que, no ISOT balanceado, o SVM permaneça à frente. Esses dois resultados orientam a configuração de referência do paradigma hiperdimensional para dados textuais, com a ressalva de que foram obtidos no conjunto de teste, sem validação separada (Seção 5.8).

6.1.1 Validação da Eficiência Energética

O ganho de eficiência energética foi a evidência mais consistente e expressiva dos experimentos. Em todos os três conjuntos de dados, os modelos HDC consumiram entre 18 e aproximadamente 1.161 vezes menos energia que o BERT no treinamento — e de 17 a 86 vezes menos na inferência —, com o OnlineHD com *Word2Vec* emergindo como configuração mais favorável (0,073 Wh contra 84,73 Wh do BERT no ISOT, com perda de 5,26 pontos percentuais de acurácia). Este resultado posiciona o HDC como tecnologia habilitadora legítima para a agenda de *Green AI* (STRUBELL; GANESH; MCCALLUM, 2019).

6.1.2 Equivalência Aproximada em Cenário Balanceado (ISOT)

No ISOT Fake News Dataset, em condição balanceada de notícias longas (22.851 *fakes* vs 21.416 reais), as melhores variantes HDC (OnlineHD e DistHD, com *Word2Vec*) alcançaram cerca de 94-95% de acurácia, contra 99,73-99,79% dos *Transformers*. Esta proximidade — um *gap* de aproximadamente 5 pontos percentuais — combinada com a redução energética de aproximadamente $1.161\times$ neste cenário valida a álgebra hiperdimensional como compressão semântica eficaz para textos jornalísticos longos.

6.1.3 Limitação no Cenário Desbalanceado (COVID-19)

No COVID-19 Fake News Dataset (4.532 reais contra 1.443 *fakes*), as melhores variantes HDC alcançaram *F1-Scores* de 67-70%, contra 82-83% dos *Transformers*. O paradigma superou os modelos de referência clássicos por margem modesta — 1,82 ponto percentual sobre o SVM no COVID-19 e 4,53 sobre a *Logistic Regression* no FEVER —, operando na mesma faixa desses classificadores. Os resultados não confirmam, portanto, a robustez intrínseca ao desbalanceamento frequentemente reivindicada na literatura HDC. A vantagem neste cenário restringe-se à eficiência operacional ($18\times$ menos energia que o BERT).

6.1.4 Limitação no Cenário de Larga Escala (FEVER)

No FEVER binarizado (109.809 instâncias), os modelos HDC atingiram *F1-Score* na faixa de 41-54%, contra 69-70% dos *Transformers*. O *gap* reflete a dependência crítica deste

conjunto de dados de conhecimento contextual externo, ausente do texto bruto das declarações. A escalabilidade computacional do HDC foi plenamente confirmada (treinamento em cerca de 120 segundos, contra 2.532 segundos do BERT, e inferência sobre as 32.943 instâncias de teste em cerca de 3,4 segundos, contra 58 segundos do BERT), mas sua eficácia preditiva neste regime é insuficiente como classificador autônomo, recomendando-se uso em arquitetura de filtragem hierárquica.

6.2 Resposta aos Objetivos da Pesquisa

Retomando os objetivos específicos delineados na Introdução:

- **OE1 — Eficácia de classificadores hiperdimensionais na detecção de *fake news*:** atingido, com ressalvas. As cinco variantes implementadas classificaram conteúdo noticioso nos experimentos conduzidos, com desempenho próximo aos *Transformers* no cenário balanceado ($gap \approx 5$ pp para as melhores variantes no ISOT) e na mesma faixa dos modelos de referência clássicos nos demais cenários. A eficiência muito superior verifica-se em relação aos *Transformers*; em relação aos classificadores clássicos, o custo do HDC é comparável ou superior (Seção 5.6).
- **OE2 — Robustez do paradigma em cenários de desbalanceamento:** parcialmente atingido. O HDC manteve-se operacional sob desbalanceamento de 3:1 e superou os modelos de referência clássicos em *F1-Score*, ainda que por margens modestas (1,82 e 4,53 pontos percentuais). A reivindicação de robustez intrínseca ao desbalanceamento, contudo, não foi confirmada empiricamente: a vantagem observada é da ordem de grandeza da variação entre classificadores clássicos.
- **OE3 — Escalabilidade no processamento de grandes volumes de dados:** plenamente atingido. O processamento de 109.809 instâncias em ordem de minutos, sem degradação de memória, confirma a complexidade linear teórica do paradigma.
- **OE4 — Quantificação do ganho de eficiência energética:** plenamente atingido. Reduções de $18\times$ a aproximadamente $1.161\times$ no treinamento e de $17\times$ a $86\times$ na inferência foram empiricamente caracterizadas, fornecendo evidência quantitativa para o argumento de sustentabilidade.

Em conjunto, o cumprimento dos quatro objetivos específicos responde ao objetivo geral formulado na Introdução: o *trade-off* entre acurácia preditiva e eficiência computacional do HDC foi investigado empiricamente e caracterizado, identificando-se os cenários em que a álgebra hiperdimensional oferece vantagem prática — sobretudo a triagem de grandes

volumes de conteúdo sob restrição de recursos — e aqueles em que o estado da arte permanece preferível.

6.3 Implicações Teóricas e Práticas

Os resultados obtidos transcendem a mera validação de um algoritmo; eles sugerem uma reavaliação dos paradigmas de eficiência na IA aplicada à segurança da informação.

Implicações teóricas. Do ponto de vista da Ciência da Computação, o sucesso do HDC no ISOT ($gap \approx 5$ pp em relação ao BERT) desafia a premissa de que compreensão semântica profunda — tipicamente associada a *Transformers* — seja pré-requisito absoluto para detecção eficaz de *fake news* em textos longos. Os dados sugerem que a “impressão digital” estatística de textos falsos pode ser capturada eficazmente em espaços vetoriais de alta dimensão através de operações algébricas simples (KANERVA, 2009). Por outro lado, os resultados no FEVER ($gap \approx 16$ pp) confirmam que a compreensão semântica profunda permanece necessária para tarefas que exigem raciocínio sobre evidências externas.

Implicações práticas (*Green AI* e *Edge Computing*). A eficiência linear $O(N \cdot L)$ demonstrada empiricamente posiciona o HDC como tecnologia habilitadora para IA Verde. Enquanto o treinamento de *Transformers* para detecção de desinformação exige GPUs de alto desempenho e consumo energético comparável ao de eletrodomésticos por horas, o HDC viabiliza *pipelines* de classificação em larga escala com consumo equivalente a uma lâmpada LED. Isso democratiza o acesso a ferramentas de verificação e permite operação local em dispositivos de borda — alinhando-se às tendências de sistemas descentralizados (JAYAKODY; MOHAMMAD; HALGAMUGE, 2022).

6.4 Limitações do Estudo

É imperativo reconhecer as limitações metodológicas e de escopo desta pesquisa:

- **Ausência de modelagem de evidências:** o HDC opera como classificador “caixa-preta” baseado em padrões textuais, não extraíndo nem validando evidências externas para justificar por que uma declaração é falsa. Isto limita sua utilidade para fact-checkers humanos que necessitam de explicabilidade (ALHINDI; PETRIDIS; MURESAN, 2018).
- **Simplificação da tarefa de verificação:** a classe NOT ENOUGH INFO do FEVER foi descartada e a tarefa foi reduzida à classificação binária do *claim*, sem recuperação

de evidências nem veredictos graduais. Detecção de meias-verdades, em que LLMs modernos mantêm vantagem semântica (HU et al., 2024), não foi avaliada.

- **Restrição unimodal:** o estudo focou exclusivamente em texto. A desinformação moderna é frequentemente multimodal (texto + imagem), e sinais adicionais — como a propagação social em rede e as características de usuários e o conhecimento externo explorados em modelos semi-supervisionados (PAKA et al., 2022) — não foram modelados.
- **Estimativa energética indireta:** o consumo foi estimado via $\text{TDP} \times \text{tempo}$, superestimando os modelos rápidos e aplicando o TDP da GPU inclusive aos modelos de referência clássicos executados em CPU. Medições diretas por NVML ou CodeCarbon forneceriam dados mais precisos. As emissões de CO_2 não foram estimadas (Seção 4.7).
- **Formulação binária do FEVER:** o uso exclusivo da partição de treinamento do FEVER, em configuração *claim-only* e binária, constitui um protocolo não-canônico da tarefa, de modo que os resultados não são diretamente comparáveis com os do FEVER *shared task* (THORNE et al., 2018) nem com *benchmarks* do LIAR (WANG, 2017; ALHINDI; PETRIDIS; MURESAN, 2018). Além disso, a deduplicação por *hash* exato não cobre quase-duplicatas (mesmo *claim* com variações de pontuação ou capitalização), o que pode introduzir viés residual; análise por similaridade fuzzy é recomendada como refinamento metodológico em trabalhos futuros.
- **Ausência de conjunto de validação:** a dimensionalidade e a codificação de referência foram definidas a partir de comparações no conjunto de teste, sem validação separada, o que pode introduzir viés otimista nos resultados do HDC (Seção 5.8).

6.5 Trabalhos Futuros

Para avançar o Estado da Arte e mitigar as limitações identificadas, sugerem-se as seguintes linhas de investigação:

- **Comparação com *Transformers* comprimidos:** estender a comparação experimental a variantes compactas da família BERT (DistilBERT, ALBERT, TinyBERT, MobileBERT), que constituem os competidores mais diretos do HDC no eixo de eficiência. Esta dissertação contrastou HDC com BERT e RoBERTa em sua forma plena; resta investigar se a vantagem energética observada se preserva quando o *baseline Transformer* é também otimizado para custo computacional reduzido.

- **Transformers como extratores de *features*:** avaliar o BERT e o RoBERTa congelados, com um classificador leve sobre suas representações, como ponto intermediário de custo entre o ajuste fino completo e o HDC; e utilizar essas mesmas representações contextuais como entrada para os codificadores hiperdimensionais.
- **Análise de estabilidade com múltiplas *seeds*:** repetir os experimentos com diferentes valores de `random_state` (por exemplo, 5 a 10 *seeds*) e reportar métricas com intervalos de confiança, refinando a robustez estatística dos resultados apresentados nesta dissertação. Recomenda-se, na mesma linha, definir a dimensionalidade e a codificação por validação cruzada no conjunto de treinamento, reservando o conjunto de teste exclusivamente para a avaliação final.
- **Extensão da grade de dimensionalidade:** avaliar os modelos HDC em dimensões superiores ($D = 5.000$, $D = 10.000$ ou mais), em infraestrutura livre das restrições de memória do ambiente utilizado nesta dissertação, caracterizando a curva de saturação entre dimensionalidade, eficácia preditiva e consumo energético.
- **Isolamento do efeito do codificador:** repetir a avaliação das cinco variantes HDC sob um codificador único e comum a todas — fixando, por exemplo, a codificação baseada em registros para o conjunto inteiro —, de modo a separar a contribuição da estratégia de atualização dos protótipos daquela do codificador adotado. Como registrado nas Seções 4.4.2 e 5.8, os codificadores padrão da biblioteca `torchhd` diferem entre variantes, o que impede essa atribuição no protocolo empregado nesta dissertação.
- **Hibridização neuro-simbólica:** arquiteturas em que a Memória de Itens do HDC é inicializada com *embeddings* pré-treinados (*Word2Vec*, BERT congelado) e refinada por *bundling*. Espera-se que esta combinação capture parte da riqueza semântica do *Deep Learning* com a eficiência algébrica do HDC.
- **Filtros hierárquicos HDC + *Transformer*:** *pipelines* em que o HDC opera como triagem de primeira linha (baixo custo), encaminhando apenas casos ambíguos para um *Transformer* (alto custo e alta precisão). Esta arquitetura permite preservar acurácia agregada com fração do custo computacional.
- **HDC para grafos de propagação:** expandir a aplicação do HDC para codificar não apenas texto, mas também topologia da rede social (nós de usuários, arestas de

compartilhamento) em hipervetores, oferecendo alternativa eficiente às *Graph Neural Networks*.

- **Implementação em *hardware* neuromórfico:** validar a eficiência energética executando os algoritmos propostos em *chips* neuromórficos (Intel Loihi, IBM TrueNorth) ou FPGAs, com medição direta de consumo em Watts.
- **Aplicação multilíngue:** adaptar os codificadores HDC para português e outras línguas, investigando o impacto da morfologia rica sobre a eficácia dos hipervetores em conjuntos de dados como o *Fake.br* Corpus.
- **Medição energética direta:** repetir os experimentos usando CodeCarbon, NVIDIA NVML ou PowerCap para substituir a estimativa indireta via TDP por medições instantâneas, oferecendo maior precisão na comparação energética.

REFERÊNCIAS

AHMED, Hadeer; TRAORE, Issa; SAAD, Sherif. Detection of online fake news using N-gram analysis and machine learning techniques. In: TRAORE, I.; AWAD, A.; WOUNGANG, I. (ed.). **Intelligent, secure, and dependable systems in distributed and cloud environments**. Cham: Springer, 2017. p. 127–138. (Lecture Notes in Computer Science, v. 10618).

ALGHAMDI, Jawaher; LUO, Suhuai; LIN, Yuqing. A comprehensive survey on machine learning approaches for fake news detection. **Multimedia Tools and Applications**, v. 83, n. 17, p. 51009–51067, 2024.

ALHINDI, Tariq; PETRIDIS, Savvas; MURESAN, Smaranda. Where is your evidence: improving fact-checking by justification modeling. In: WORKSHOP ON FACT EXTRACTION AND VERIFICATION (FEVER), 1., 2018, Brussels. **Proceedings [...]**. Brussels: Association for Computational Linguistics, 2018. p. 85–90.

DEVLIN, Jacob; CHANG, Ming-Wei; LEE, Kenton; TOUTANOVA, Kristina. BERT: pre-training of deep bidirectional transformers for language understanding. In: CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES (NAACL-HLT), 2019, Minneapolis. **Proceedings [...]**. Minneapolis: ACL, 2019. p. 4171–4186.

DUAN, Shijin et al. LeHDC: learning-based hyperdimensional computing classifier. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE (DAC), 59., 2022, San Francisco. **Proceedings [...]**. San Francisco: ACM/IEEE, 2022. p. 1111–1116.

GUNDAPU, Sunil; MAMIDI, Radhika. **Transformer based automatic COVID-19 fake news detection system**. arXiv:2101.00180, 2021. Preprint. Disponível em: <https://arxiv.org/abs/2101.00180>. Acesso em: 1 abr. 2026.

HAMED, Suhaib Kh.; AB AZIZ, Mohd Juzaidin; YAAKUB, Mohd Ridzwan. A review of fake news detection approaches: a critical analysis of relevant studies and highlighting key challenges associated with the dataset, feature representation, and data fusion. **Heliyon**, v. 9, n. 10, e20382, 2023.

HARRIS, Zellig S. Distributional structure. **Word**, v. 10, n. 2-3, p. 146–162, 1954.

HEDDES, Mike; NUNES, Igor; VERGÉS, Pere; KLEYKO, Denis; ABRAHAM, Danny; GIVARGIS, Tony; NICOLAU, Alexandru; VEIDENBAUM, Alexander. Torchhd: an open source Python library to support research on hyperdimensional computing and vector symbolic architectures. **Journal of Machine Learning Research**, v. 24, n. 255, p. 1–10, 2023.

HERNÁNDEZ-CANO, Alejandro; MATSUMOTO, Namiko; PING, Eric; IMANI, Mohsen. OnlineHD: robust, efficient, and single-pass online learning using hyperdimensional system. In: DESIGN, AUTOMATION AND TEST IN EUROPE CONFERENCE (DATE), 2021, Grenoble. **Proceedings** [...]. Grenoble: IEEE, 2021. p. 56–61.

HU, Beizhe et al. Bad actor, good advisor: exploring the role of large language models in fake news detection. **Proceedings of the AAAI Conference on Artificial Intelligence**, v. 38, n. 20, p. 22105–22113, 2024.

IMANI, Mohsen et al. AdapthD: adaptive efficient training for brain-inspired hyperdimensional computing. In: IEEE BIOMEDICAL CIRCUITS AND SYSTEMS CONFERENCE (BioCAS), 2019, Nara. **Proceedings** [...]. Nara: IEEE, 2019a. p. 1–4.

IMANI, Mohsen et al. QuantHD: a quantization framework for hyperdimensional computing. **IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems**, v. 39, n. 10, p. 2268–2278, 2019b.

IMANI, Mohsen et al. CompHD: efficient hyperdimensional computing using model compression. In: IEEE/ACM INTERNATIONAL SYMPOSIUM ON LOW POWER ELECTRONICS AND DESIGN (ISLPED), 2019, Lausanne. **Proceedings** [...]. Lausanne: IEEE, 2019c. p. 1–6.

JAYAKODY, Nirosh; MOHAMMAD, Azeem; HALGAMUGE, Malka N. Fake news detection using a decentralized deep learning model and federated learning. In: ANNUAL CONFERENCE OF THE IEEE INDUSTRIAL ELECTRONICS SOCIETY (IECON), 48., 2022, Brussels. **Proceedings** [...]. Brussels: IEEE, 2022. p. 1–6.

JOSHI, Aditya; HALSETH, Johan T.; KANERVA, Pentti. Language geometry using random indexing. In: QUANTUM INTERACTION (QI 2016), 2016. **Proceedings** [...]. Cham: Springer, 2017. p. 265–274. (Lecture Notes in Computer Science, v. 10106).

KALIYAR, Rohit Kumar; GOSWAMI, Anurag; NARANG, Pratik. FakeBERT: fake news detection in social media with a BERT-based deep learning approach. **Multimedia Tools and Applications**, v. 80, n. 8, p. 11765–11788, 2021.

KANERVA, Pentti. **Sparse distributed memory**. Cambridge: MIT Press, 1988.

KANERVA, Pentti. Hyperdimensional computing: an introduction to computing in distributed representation with high-dimensional random vectors. **Cognitive Computation**, v. 1, n. 2, p. 139–159, 2009.

KIM, Yeseong; IMANI, Mohsen; ROSING, Tajana. BinHD: encoding hyperdimensional vectors with binary operations for efficient learning. In: IEEE/ACM INTERNATIONAL CONFERENCE ON COMPUTER-AIDED DESIGN (ICCAD), 2018, San Diego. **Proceedings** [...]. San Diego: IEEE, 2018. p. 1–8.

KIM, Yoon. Convolutional neural networks for sentence classification. In: CONFERENCE ON EMPIRICAL METHODS IN NATURAL LANGUAGE PROCESSING (EMNLP), 2014, Doha. **Proceedings** [...]. Doha: ACL, 2014. p. 1746–1751.

KLEYKO, Denis; RACHKOVSKIJ, Dmitri A.; OSIPOV, Evgeny; RAHIMI, Abbas. A survey on hyperdimensional computing aka vector symbolic architectures, part I: models and data transformations. **ACM Computing Surveys**, v. 55, n. 6, art. 130, p. 1–40, 2023a.

KLEYKO, Denis; RACHKOVSKIJ, Dmitri A.; OSIPOV, Evgeny; RAHIMI, Abbas. A survey on hyperdimensional computing aka vector symbolic architectures, part II: applications, cognitive models, and challenges. **ACM Computing Surveys**, v. 55, n. 9, art. 175, p. 1–52, 2023b.

LAZER, David M. J. et al. The science of fake news. **Science**, v. 359, n. 6380, p. 1094–1096, 2018.

LIU, Yinhan et al. **RoBERTa**: a robustly optimized BERT pretraining approach. arXiv:1907.11692, 2019. Preprint. Disponível em: <https://arxiv.org/abs/1907.11692>. Acesso em: 1 abr. 2026.

MIKOLOV, Tomas; CHEN, Kai; CORRADO, Greg; DEAN, Jeffrey. Efficient estimation of word representations in vector space. In: INTERNATIONAL CONFERENCE ON LEARNING REPRESENTATIONS (ICLR), 2013, Scottsdale. **Workshop track proceedings** [...]. Scottsdale: ICLR, 2013. p. 1–12.

OLAN, Femi; JAYAWICKRAMA, Uchitha; ARAKPOGUN, Emmanuel Ogiemwonyi; SUKLAN, Jana; LIU, Shaofeng. Fake news on social media: the impact on society. **Information Systems Frontiers**, v. 26, n. 2, p. 443–458, 2024.

PAKA, William Scott et al. Cross-SEAN: a cross-stitch semi-supervised neural attention model for COVID-19 fake news detection. **Applied Soft Computing**, v. 107, p. 107393, 2022.

PATWA, Parth et al. Fighting an infodemic: COVID-19 fake news dataset. In: CHAKRABORTY, T. et al. (ed.). **Combating online hostile posts in regional languages during emergency situation**: CONSTRAINT 2021. Cham: Springer, 2021. p. 21–29. (Communications in Computer and Information Science, v. 1402).

PLATE, Tony A. **Holographic reduced representation**: distributed representation for cognitive structures. Stanford: CSLI Publications, 2003.

RUCHANSKY, Natali; SEO, Sungyong; LIU, Yan. CSI: a hybrid deep model for fake news detection. In: ACM ON CONFERENCE ON INFORMATION AND KNOWLEDGE MANAGEMENT (CIKM), 2017, Singapore. **Proceedings** [...]. Singapore: ACM, 2017. p. 797–806.

SALAMAT, Sahand; IMANI, Mohsen; ROSING, Tajana. SparseHD: algorithm-hardware co-optimization for efficient high-dimensional computing. In: IEEE INTERNATIONAL SYMPOSIUM ON FIELD-PROGRAMMABLE CUSTOM COMPUTING MACHINES (FCCM), 27., 2019, San Diego. **Proceedings** [...]. San Diego: IEEE, 2019. p. 190–198.

SHAHI, Gautam Kishore; DIRKSON, Anne; MAJCHRZAK, Tim A. An exploratory study of COVID-19 misinformation on Twitter. **Online Social Networks and Media**, v. 22, p. 100104, 2022.

SHU, Kai et al. Fake news detection on social media: a data mining perspective. **ACM SIGKDD Explorations Newsletter**, v. 19, n. 1, p. 22–36, 2017.

SINGH, Mohit Kumar; AHMED, Jawed; ALAM, Mohammad Afshar; RAGHUVANSHI, Kamal Kumar; KUMAR, Sushil. A comprehensive review on automatic detection of fake news on social media. **Multimedia Tools and Applications**, v. 83, n. 16, p. 47319–47352, 2024.

STRUBELL, Emma; GANESH, Ananya; MCCALLUM, Andrew. Energy and policy considerations for deep learning in NLP. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL), 57., 2019, Florence. **Proceedings** [...]. Florence: ACL, 2019. p. 3645–3650.

THORNE, James; VLACHOS, Andreas; CHRISTODOULOPOULOS, Christos; MITTAL, Arpit. FEVER: a large-scale dataset for Fact Extraction and VERification. In: CONFERENCE OF THE NORTH AMERICAN CHAPTER OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS: HUMAN LANGUAGE TECHNOLOGIES (NAACL-HLT), 2018, New Orleans. **Proceedings** [...]. New Orleans: Association for Computational Linguistics, 2018. p. 809–819.

VASWANI, Ashish et al. Attention is all you need. In: CONFERENCE ON NEURAL INFORMATION PROCESSING SYSTEMS (NeurIPS), 2017, Long Beach. **Advances in neural information processing systems**, v. 30. Long Beach: NeurIPS, 2017. p. 5998–6008.

VOSOUGHI, Soroush; ROY, Deb; ARAL, Sinan. The spread of true and false news online. **Science**, v. 359, n. 6380, p. 1146–1151, 2018.

WANG, Junyao; HUANG, Sitao; IMANI, Mohsen. DistHD: a learner-aware dynamic encoding method for hyperdimensional classification. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE (DAC), 60., 2023, San Francisco. **Proceedings** [...]. San Francisco: IEEE, 2023. p. 1–6.

WANG, Ruixuan; JIAO, Xun; HU, X. Sharon. ODHD: one-class brain-inspired hyperdimensional computing for outlier detection. In: ACM/IEEE DESIGN AUTOMATION CONFERENCE (DAC), 59., 2022, San Francisco. **Proceedings** [...]. San Francisco: ACM, 2022. p. 43–48.

WANG, William Yang. “Liar, liar pants on fire”: a new benchmark dataset for fake news detection. In: ANNUAL MEETING OF THE ASSOCIATION FOR COMPUTATIONAL LINGUISTICS (ACL), 55., 2017, Vancouver. **Proceedings** [...]. Vancouver: ACL, 2017. p. 422–426.

WARDLE, Claire; DERAKHSHAN, Hossein. **Information disorder**: toward an interdisciplinary framework for research and policy making. Strasbourg: Council of Europe, 2017.

XU, Wenrui et al. IFODHD: improved feature selection based outlier detection using hyperdimensional computing. **Journal of Signal Processing Systems**, v. 96, n. 12, p. 791–807, 2025.

ZOU, Zhihao et al. Scalable edge-based hyperdimensional learning system with brain-like neural adaptation. In: INTERNATIONAL CONFERENCE FOR HIGH PERFORMANCE COMPUTING, NETWORKING, STORAGE AND ANALYSIS (SC), 2021, St. Louis. **Proceedings** [...]. St. Louis: ACM, 2021. p. 1–15.

APÊNDICE A – RESULTADOS COMPLETOS POR EXPERIMENTO

Este apêndice reporta os valores brutos da matriz de confusão --- Verdadeiros Positivos (TP), Verdadeiros Negativos (TN), Falsos Positivos (FP) e Falsos Negativos (FN), com a classe *fake* definida como positiva --- para todas as 96 configurações avaliadas (modelo $\times$ codificação $\times$ conjunto de dados). Os modelos HDC são reportados na dimensão adotada no estudo ($D = 3.000$); os modelos clássicos e os *Transformers* não dependem de dimensionalidade hiperdimensional. Diferentemente das tabelas do Capítulo 5, que apresentam as métricas já calculadas, e das Figuras 8 a 13, que ilustram graficamente apenas o subconjunto em *Bag of Words*, esta tabela disponibiliza a contagem completa que origina todas as métricas, permitindo a qualquer leitor recalculá-las: Acurácia, Precisão, Revocação, *F1-Score* e derivados, bem como auditar o comportamento de cada classificador sob desbalanceamento. Cabe uma nota de leitura: embora todos os *pipelines* adotem o mesmo protocolo de particionamento (70/30, `random_state = 42`), a execução independente do particionamento em cada *pipeline* de codificação produz pequenas variações na contagem de instâncias por classe no conjunto de teste — observáveis nas somas $TP + FN$ entre linhas de um mesmo conjunto de dados —, permanecendo idêntico o total de instâncias de teste de cada corpus.

Tabela A.1 – Matrizes de Confusão de Todas as Configurações (classe positiva = *fake*)

Dataset	Modelo	Codif.	TP	TN	FP	FN
ISOT	<i>Vanilla</i> (HDC, $D=3.000$)	TF-IDF	4.663	4.450	1.976	2.192
	<i>Vanilla</i> (HDC, $D=3.000$)	BoW	5.476	5.724	701	1.380
	<i>Vanilla</i> (HDC, $D=3.000$)	Word2Vec	5.447	5.619	834	1.381
	AdaptHD (HDC, $D=3.000$)	TF-IDF	4.337	3.529	2.897	2.518
	AdaptHD (HDC, $D=3.000$)	BoW	6.677	6.015	410	179
	AdaptHD (HDC, $D=3.000$)	Word2Vec	6.818	719	5.734	10
	OnlineHD (HDC, $D=3.000$)	TF-IDF	6.249	5.851	575	606
	OnlineHD (HDC, $D=3.000$)	BoW	5.957	5.373	1.052	899
	OnlineHD (HDC, $D=3.000$)	Word2Vec	6.469	6.085	368	359
	NeuralHD (HDC, $D=3.000$)	TF-IDF	6.106	6.047	379	749
	NeuralHD (HDC, $D=3.000$)	BoW	5.349	5.145	1.280	1.507
	NeuralHD (HDC, $D=3.000$)	Word2Vec	5.923	5.835	618	905
	DistHD (HDC, $D=3.000$)	TF-IDF	6.238	5.886	540	617
	DistHD (HDC, $D=3.000$)	BoW	6.399	6.143	282	457
	DistHD (HDC, $D=3.000$)	Word2Vec	6.464	6.091	362	364
	<i>Logistic Regression</i>	TF-IDF	6.489	5.958	366	468
	<i>Logistic Regression</i>	BoW	6.602	6.142	182	355
	<i>Logistic Regression</i>	Word2Vec	6.492	6.060	366	363
	<i>Random Forest</i>	TF-IDF	6.720	6.074	250	237
	<i>Random Forest</i>	BoW	6.717	6.050	274	240
	<i>Random Forest</i>	Word2Vec	6.525	6.076	350	330

Dataset	Modelo	Codif.	TP	TN	FP	FN
	SVM	TF-IDF	6.537	6.075	249	420
	SVM	BoW	5.470	5.167	1.157	1.487
	SVM	Word2Vec	6.555	6.132	294	300
	Árvore de Decisão	TF-IDF	6.809	6.186	138	148
	Árvore de Decisão	BoW	6.804	6.200	124	153
	Árvore de Decisão	Word2Vec	6.481	5.991	435	374
	KNN	TF-IDF	5.891	4.885	1.439	1.066
	KNN	BoW	5.015	5.103	1.221	1.942
	KNN	Word2Vec	6.432	6.012	414	423
	BERT	—	6.842	6.411	15	13
	RoBERTa	—	6.837	6.408	18	18
COVID-19	Vanilla (HDC, D=3.000)	TF-IDF	308	1.190	177	118
	Vanilla (HDC, D=3.000)	BoW	308	1.048	312	125
	Vanilla (HDC, D=3.000)	Word2Vec	323	958	394	118
	AdaptHD (HDC, D=3.000)	TF-IDF	268	1.243	124	158
	AdaptHD (HDC, D=3.000)	BoW	355	735	625	78
	AdaptHD (HDC, D=3.000)	Word2Vec	215	1.249	103	226
	OnlineHD (HDC, D=3.000)	TF-IDF	331	1.218	149	95
	OnlineHD (HDC, D=3.000)	BoW	301	1.205	155	132
	OnlineHD (HDC, D=3.000)	Word2Vec	298	1.240	112	143
	NeuralHD (HDC, D=3.000)	TF-IDF	330	1.221	146	96
	NeuralHD (HDC, D=3.000)	BoW	341	1.020	340	92
	NeuralHD (HDC, D=3.000)	Word2Vec	356	1.083	269	85
	DistHD (HDC, D=3.000)	TF-IDF	314	1.219	148	112
	DistHD (HDC, D=3.000)	BoW	303	1.158	202	130
	DistHD (HDC, D=3.000)	Word2Vec	284	1.228	124	157
	Logistic Regression	TF-IDF	45	1.362	5	381
	Logistic Regression	BoW	220	1.322	45	206
	Logistic Regression	Word2Vec	248	1.297	48	200
	Random Forest	TF-IDF	104	1.355	12	322
	Random Forest	BoW	100	1.351	16	326
	Random Forest	Word2Vec	214	1.315	30	234
	SVM	TF-IDF	56	1.362	5	370
	SVM	BoW	201	1.331	36	225
	SVM	Word2Vec	249	1.312	33	199
	Árvore de Decisão	TF-IDF	238	1.241	126	188
	Árvore de Decisão	BoW	241	1.226	141	185
	Árvore de Decisão	Word2Vec	245	1.266	79	203
	KNN	TF-IDF	201	1.291	76	225
	KNN	BoW	28	1.358	9	398
	KNN	Word2Vec	277	1.184	161	171
	BERT	—	380	1.260	85	68
	RoBERTa	—	358	1.273	72	90
FEVER	Vanilla (HDC, D=3.000)	TF-IDF	4.624	16.135	7.867	4.317
	Vanilla (HDC, D=3.000)	BoW	5.319	15.734	8.276	3.614
	Vanilla (HDC, D=3.000)	Word2Vec	5.498	13.246	10.749	3.450
	AdaptHD (HDC, D=3.000)	TF-IDF	2.705	18.342	5.660	6.236
	AdaptHD (HDC, D=3.000)	BoW	6.665	10.902	13.108	2.268
	AdaptHD (HDC, D=3.000)	Word2Vec	5.290	14.673	9.322	3.658
	OnlineHD (HDC, D=3.000)	TF-IDF	3.585	18.453	5.549	5.356
	OnlineHD (HDC, D=3.000)	BoW	3.673	22.006	2.004	5.260
	OnlineHD (HDC, D=3.000)	Word2Vec	4.481	17.302	6.693	4.467
	NeuralHD (HDC, D=3.000)	TF-IDF	3.760	19.303	4.699	5.181
	NeuralHD (HDC, D=3.000)	BoW	3.831	21.065	2.945	5.102

Dataset	Modelo	Codif.	TP	TN	FP	FN
	NeuralHD (HDC, D=3.000)	Word2Vec	5.393	18.222	5.773	3.555
	DistHD (HDC, D=3.000)	TF-IDF	3.662	18.290	5.712	5.279
	DistHD (HDC, D=3.000)	BoW	3.803	21.601	2.409	5.130
	DistHD (HDC, D=3.000)	Word2Vec	3.761	18.416	5.579	5.187
	<i>Logistic Regression</i>	TF-IDF	228	23.868	20	8.827
	<i>Logistic Regression</i>	BoW	2.372	22.935	953	6.683
	<i>Logistic Regression</i>	Word2Vec	3.291	22.825	1.108	5.719
	<i>Random Forest</i>	TF-IDF	2.024	23.467	421	7.031
	<i>Random Forest</i>	BoW	2.180	23.457	431	6.875
	<i>Random Forest</i>	Word2Vec	2.420	23.079	855	6.589
	SVM	TF-IDF	241	23.853	35	8.814
	SVM	BoW	1.154	23.756	132	7.901
	SVM	Word2Vec	2.846	23.198	735	6.164
	Árvore de Decisão	TF-IDF	3.504	20.744	3.144	5.551
	Árvore de Decisão	BoW	3.446	21.319	2.569	5.609
	Árvore de Decisão	Word2Vec	3.025	22.048	1.886	5.984
	KNN	TF-IDF	2.252	21.585	2.303	6.803
	KNN	BoW	1.210	22.204	1.684	7.845
	KNN	Word2Vec	2.869	22.020	1.913	6.141
	BERT	—	5.407	22.921	1.012	3.603
	RoBERTa	—	5.438	22.699	1.234	3.572

Fonte: Elaborado pelo autor (2026).

Modelos HDC em D = 3.000; classe positiva = fake.

Tabela A.2 – Matrizes de Confusão em D = 1.000 sob Codificação TF-IDF

Dataset	Modelo	Codif.	TP	TN	FP	FN
ISOT	<i>Vanilla</i> (HDC, D=1.000)	TF-IDF	4.178	3.914	2.512	2.677
ISOT	AdaptHD (HDC, D=1.000)	TF-IDF	3.473	3.721	2.705	3.382
ISOT	OnlineHD (HDC, D=1.000)	TF-IDF	6.148	5.719	707	707
ISOT	NeuralHD (HDC, D=1.000)	TF-IDF	6.076	5.978	448	779
ISOT	DistHD (HDC, D=1.000)	TF-IDF	6.246	5.793	633	609
COVID-19	<i>Vanilla</i> (HDC, D=1.000)	TF-IDF	294	1.097	270	132
COVID-19	AdaptHD (HDC, D=1.000)	TF-IDF	283	1.264	103	143
COVID-19	OnlineHD (HDC, D=1.000)	TF-IDF	324	1.223	144	102
COVID-19	NeuralHD (HDC, D=1.000)	TF-IDF	331	1.210	157	95
COVID-19	DistHD (HDC, D=1.000)	TF-IDF	330	1.223	144	96
FEVER	<i>Vanilla</i> (HDC, D=1.000)	TF-IDF	4.706	15.157	8.845	4.235
FEVER	AdaptHD (HDC, D=1.000)	TF-IDF	2.562	18.371	5.631	6.379
FEVER	OnlineHD (HDC, D=1.000)	TF-IDF	3.576	18.264	5.738	5.365
FEVER	NeuralHD (HDC, D=1.000)	TF-IDF	3.791	19.110	4.892	5.150
FEVER	DistHD (HDC, D=1.000)	TF-IDF	3.673	18.218	5.784	5.268

Fonte: Elaborado pelo autor (2026).

Modelos HDC em D = 1.000; classe positiva = fake.