Uma Análise Comparativa de Algoritmos de Machine Learning e Explicabilidade para Detecção de Intrusão de Redes

Jhonatas G. Ribeiro[1], **Igor B. Reis**[1], **Jurandir C. Lacerda Jr.**[1]

[1]Instituto Federal de Educação, Ciência e Tecnologia do Piauí (IFPI)
Corrente, Piauí, Brasil

jhonatasgomes2003@gmail.com
{igor.bezerra,jurandir.cavalcante}@ifpi.edu.br

Abstract. *In high-throughput networks, where milliseconds separate detection from data exfiltration, precision is paramount. This work analyzes the performance of six machine learning algorithms on the CIC-IDS2017 dataset, addressing the dilemma between the "Black Box" and the need for real-time response. Through a complete pipeline, a decisive trade-off was revealed: while Random Forest achieves an accuracy of 99.64%, CatBoost emerges as the superior choice for active defense, delivering inference* $3.4\times$ *faster (0.65 s) with negligible accuracy loss. Finally, Explainable AI techniques were applied to interpret model decisions based on legitimate network patterns, transforming alerts into auditable security.*

Resumo. *Em redes de alta vazão, onde milissegundos separam a detecção da exfiltração de dados, a precisão é fundamental. Este trabalho analisa o desempenho de seis algoritmos de aprendizado de máquina no dataset CIC-IDS2017, enfrentando o dilema entre a "Caixa Preta" e a necessidade de resposta em tempo real. Através de um pipeline completo, foi revelado um trade-off decisivo: enquanto o Random Forest apresenta a precisão de 99,64%, o CatBoost emerge como a escolha superior para defesa ativa, entregando inferência* $3,4\times$ *mais rápida (0,65 s) com perda desprezível de acurácia. Por fim, aplicaram-se técnicas de IA Explicável para interpretar a decisão dos modelos baseando-se em padrões de rede legítimos e podendo transformar alertas em segurança auditável.*

1. Introdução

A arquitetura das redes de computadores modernas enfrenta um cenário de ameaças sem precedentes, caracterizado pela crescente sofisticação dos vetores de ataque e pelo aumento exponencial no volume de tráfego de dados (Anis et al., 2025; Arreche et al., 2024). A massiva adoção de protocolos criptografados, que inviabilizam a inspeção profunda de conteúdo, somada à vertiginosa velocidade de surgimento de novas variantes de *malware*, tornaram os Sistemas de Detecção de Intrusão (IDS) baseados em assinaturas estáticas, que operam comparando padrões de bytes conhecidos contra um banco de dados de ameaças pré-mapeadas, progressivamente ineficazes (Elasaad et al., 2025). A incapacidade desses sistemas de generalizar padrões desconhecidos forçou a indústria e a academia a adotarem abordagens preditivas baseadas em Aprendizado de Máquina (*Machine Learning* - ML), capazes de identificar anomalias estatísticas sutis em fluxos de rede complexos.

Fundamentalmente, os IDS atuam como sentinelas digitais, monitorando o tráfego de rede em busca de atividades suspeitas. Tradicionalmente, eles são categorizados em duas vertentes: baseados em assinatura, que buscam padrões conhecidos de ameaças; e baseados em anomalia, que modelam o comportamento normal da rede para identificar desvios. O cenário de ataques atual é vasto, variando desde varreduras de portas (*PortScan*) e tentativas de força bruta, até ataques volumétricos de Negação de Serviço (DoS/DDoS) e explorações sofisticadas de camada de aplicação (*Web Attacks*) e infiltrações furtivas (*Botnets*), exigindo que os IDSs modernos sejam capazes de generalizar a detecção para além de assinaturas estáticas.

No entanto, a transição de sistemas determinísticos para modelos probabilísticos de ML introduz dois desafios críticos que dificultam sua implantação em ambientes de produção. O primeiro é a opacidade decisória, conhecida como o problema da "Caixa Preta". Modelos de alta performance, como Redes Neurais de Aprendizado Profundo (*Deep Learning* - DL), operam frequentemente como oráculos matemáticos complexos, falhando em fornecer justificativas inteligíveis para seus alertas. Conforme destacado por Wali et al. (2025), essa falta de transparência gera uma crise de confiança entre analistas de segurança, que hesitam em automatizar respostas a incidentes sem compreender a causa raiz, temendo que falsos positivos interrompam operações críticas de negócio.

O segundo desafio reside na eficiência computacional necessária para operar em redes de alta vazão. Embora arquiteturas de DL tenham ganhado popularidade, elas frequentemente exigem um custo computacional proibitivo para inferência em tempo real, criando gargalos de latência inaceitáveis para a defesa ativa de redes corporativas e *backbones* (Anis et al., 2025). Khan et al. (2024b) apontam que a complexidade excessiva dos modelos pode inviabilizar a escalabilidade do IDS, criando um impasse técnico: como manter a alta precisão na detecção de ataques modernos sem comprometer a performance da rede com processamento pesado?

Para tratar simultaneamente as lacunas de interpretabilidade e eficiência, este trabalho fornece um *pipeline* de detecção de intrusão otimizado, focado no equilíbrio entre acurácia, velocidade de inferência e explicabilidade. A abordagem deste trabalho prioriza algoritmos que oferecem robustez sem a complexidade desnecessária de redes neurais profundas.

As contribuições deste artigo são tripartites. Primeiramente, foi realizada uma avaliação comparativa rigorosa de seis algoritmos de ML no *dataset* CIC-IDS2017, demonstrando que métodos de *Ensemble* baseados em árvores (*Random Forest* e *Gradient Boosting*), quando aliados a um pré-processamento com seleção de características (*SelectKBest*) e balanceamento (*SMOTE*), superam ou igualam a performance de abordagens de DL da literatura, validando a tese de que dados tabulares de rede são melhor modelados por particionamento hierárquico (Grinsztajn et al., 2022).

Em segundo lugar, foi apresentada uma análise crítica de *trade-off* operacional. Foi evidenciado que, enquanto o *Random Forest* maximiza a acurácia global, o modelo *CatBoost* oferece uma alternativa superior para monitoramento em tempo real, entregando uma velocidade de inferência maior com perda desprezível de precisão, viabilizando sua aplicação em redes de alta velocidade. Por fim, foi integrado o *framework* de Inteligência Artificial Explicável (*explainable artificial intelligence* - XAI) via *SHapley Additive exPla-*

nations (SHAP) para auditar o modelo, fornecendo visualizações que permitem validar tecnicamente padrões aprendidos, como a correlação entre variância de pacotes e ataques volumétricos, aumentando a transparência e a auditabilidade do sistema de defesa.

O restante deste artigo está organizado da seguinte forma: a Seção 2 apresenta uma revisão bibliográfica detalhada sobre os trabalhos relacionados e o estado da arte em sistemas de detecção de intrusão baseados em inteligência artificial. A Seção 3 descreve a metodologia experimental adotada, detalhando as características do *dataset* CIC-IDS2017, os procedimentos de limpeza de dados, engenharia de atributos e o ambiente de hardware utilizado. Na Seção 4, são apresentados e discutidos os resultados obtidos, focando no *trade-off* entre acurácia e tempo de inferência, além da auditoria técnica via XAI. Por fim, a Seção 5 sintetiza as conclusões extraídas deste estudo e aponta as principais diretrizes para trabalhos futuros.

2. Trabalhos Relacionados

A crescente adoção de técnicas de ML e DL para o desenvolvimento dos IDS tem representado um avanço na cibersegurança. A capacidade desses modelos de aprender padrões complexos possibilita a identificação de ameaças sofisticadas que escapam às assinaturas estáticas. No entanto, a literatura recente aponta que a transição para modelos probabilísticos introduz desafios críticos, com o elevado custo computacional e a falta de interpretabilidade, conhecida como o problema da "caixa preta".

Uma das críticas mais recorrentes aos modelos de DL é a sobrecarga operacional. Hakami et al. (2025) observaram que arquiteturas complexas como *Long Short-Term Memory* (LSTM) lutam para processar grandes volumes de dados em tempo hábil. Em sua análise comparativa, a disparidade é evidente: enquanto um modelo Random Forest exigiu apenas 92 segundos para treinamento e alcançou um tempo de inferência de 3 ms por amostra, o modelo LSTM demandou 37 minutos para treinamento e apresentou uma latência de 18 ms, seis vezes maior. Corroborando essa visão, Schmidt et al. (2025) analisaram os *trade-offs* de energia e desempenho, destacando que arquiteturas profundas frequentemente impõem um custo computacional proibitivo para aplicações de borda, assim como apontado por Anis et al. (2025) em relação a *Transformers*. Além da performance, Khan et al. (2024a) argumentam que a opacidade desses modelos mina a confiança dos analistas de segurança, pois a falta de *insight* sobre o processo decisório dificulta a distinção entre um ataque real e um falso positivo.

Para mitigar esses problemas, a comunidade científica, incluindo trabalhos recentes do SBRC 2025 (Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos), tem investigado estratégias de otimização de pré-processamento e o uso de modelos de *Ensemble*. Santos and Miani (2025) analisaram o impacto da redução de dimensionalidade, comparando PCA e seleção de atributos via Qui-quadrado, concluindo que a simplificação dos dados é vital para a generalização. Outras abordagens incluem o uso de *Extra Trees* para seleção de características (Nigar and Mustafa, 2025) e métodos híbridos combinando ANOVA e SHAP (Khan et al., 2024b). Contudo, ainda falta um consenso sobre o *pipeline* ideal que equilibre a redução de ruído com a preservação de padrões de ataques minoritários.

No campo dos classificadores, os modelos de *Ensemble* consolidaram-se como uma alternativa robusta ao DL para dados tabulares. Wali et al. (2025) propuseram

um *framework* de IDS confiável utilizando *Random Forest*, integrando um módulo de credibilidade baseado em XAI. Apesar de inovador, o estudo limita-se a um único algoritmo, não explorando o potencial de métodos de *Gradient Boosting* modernos. Paralelamente, Nigar and Mustafa (2025) aplicaram técnicas de balanceamento SMOTE para melhorar a detecção, mas não apresentaram métricas explícitas de latência de inferência, deixando uma lacuna sobre a viabilidade de sua solução em redes de alta vazão. Já Siganos et al. (2022) avançaram na auditoria de decisões utilizando *SHAP Waterfall Plots* para explicar ataques em *Internet of Things*, mas a aplicação foi restrita a cenários específicos.

A análise crítica do estado da arte revela, portanto, uma fragmentação metodológica. Identifica-se a ausência de um trabalho que integre holisticamente os componentes essenciais para um IDS de produção. A literatura atual carece de um estudo que realize simultaneamente: (i) uma comparação rigorosa entre os quatro principais *ensembles* (Random Forest, XGBoost, LightGBM e CatBoost); (ii) a aplicação de um *pipeline* de pré-processamento padronizado com SMOTE e *SelectKBest*; (iii) a avaliação explícita da latência de inferência (em milissegundos) para validar a operação em tempo real; e (iv) o uso de XAI local para auditoria técnica de alertas. A presente proposta visa preencher essa lacuna multifacetada, oferecendo uma avaliação unificada que prioriza não apenas a acurácia, mas a eficiência e a transparência.

3. Metodologia

A estratégia metodológica adotada neste trabalho foi estruturada em três etapas sequenciais para garantir a reprodutibilidade e a robustez dos resultados. Inicialmente, a Subseção 3.1 apresenta o *dataset* CIC-IDS2017 e a caracterização das ameaças, justificando a escolha deste. Na sequência, a Subseção 3.2 descreve o *pipeline* de tratamento de dados, detalhando as técnicas de seleção de características e balanceamento de classes. Por fim, a Subseção 3.3 define o ambiente experimental, especificando os algoritmos de aprendizado de máquina avaliados e a aplicação do *framework* de XAI para a auditoria das decisões.

3.1. Dataset CIC-IDS2017 e Caracterização de Ameaças

A validação experimental fundamenta-se no *dataset* CIC-IDS2017 (Sharafaldin et al., 2018), selecionado por mitigar a obsolescência de *benchmarks* pretéritos (e.x., *KDDCup99, NSL-KDD*) incapazes de representar tráfego criptografado (HTTPS) e ataques modernos. O CIC-IDS2017 foi escolhido por garantir validade estatística e comparabilidade com o estado da arte. Gerado em um *testbed* isolado durante cinco dias, o conjunto totaliza 2,8 milhões de fluxos derivados de perfis comportamentais distintos: o *B-Profile*, simulando 25 usuários em atividades naturais (*HTTP, HTTPS, FTP, SSH, e-mail*), e o *M-Profile*, executando cenários de ataque com ferramentas padrão da indústria. A organização temporal dos dados evolui em complexidade para permitir a análise de anomalias contextuais: a rotina inicia-se na segunda-feira (*baseline* benigno), progredindo para Força Bruta (terça-feira), *DoS* e *Heartbleed* (quarta-feira), ataques Web e Infiltração (quinta-feira), culminando em DDoS volumétricos e *PortScan* via *Botnets* na sexta-feira. A escolha deste *dataset* cumpre critérios críticos de validação, diversidade de protocolos e ataques viabilizando o uso de modelos operando sobre metadados.

3.2. Pré-processamento e Engenharia de Características

O *pipeline* de tratamento de dados iniciou-se pela higienização de instâncias com valores nulos ou infinitos e normalização da nomenclatura para *snake_case*. A variável alvo foi

codificada via *Label Encoding*, mantendo-se a escala original dos atributos numéricos dada a invariância característica de modelos baseados em árvore. Para tratar a dispersão da alta granularidade original (15 classes), aplicou-se um reagrupamento taxonômico em 6 macrocategorias (Tabela 1) unindo ataques de características semelhantes, simplificando a fronteira de decisão conforme ilustrado nas Figuras 1 e 2

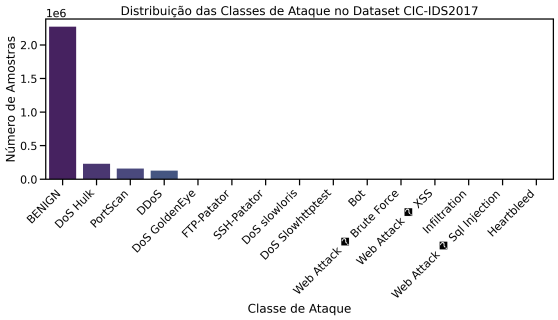


Figura 1. Distribuição original.

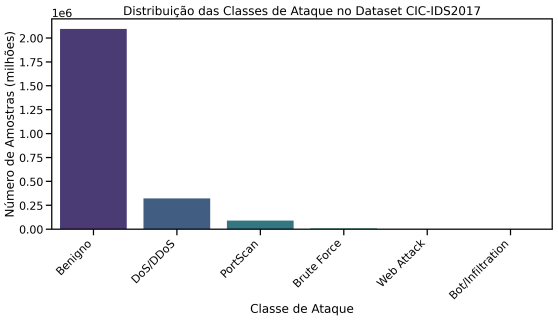


Figura 2. Distribuição agrupada.

Simultaneamente, para mitigar o custo computacional e o risco de *overfitting* inerentes à alta dimensionalidade, empregou-se filtragem univariada via algoritmo *SelectKBest* com a métrica *ANOVA F-value*. O teste quantifica a separabilidade entre classes pela razão $F = \frac{\text{Variância entre grupos}}{\text{Variância intra-grupos}}$, onde valores elevados denotam alto poder discriminativo. Para determinar o número ideal de atributos (k), conduziu-se uma análise de sensibilidade variando $k \in \{10, 20, 30, 40, 50\}$. Observou-se que a inclusão de atributos além de $k = 30$ resultava em ganho marginal de performance, porém aumentava linearmente o tempo de treinamento. Portanto, fixou-se $k = 30$ como o ponto de inflexão ótimo no compromisso entre eficiência e eficácia, reduzindo a dimensionalidade de aproximadamente 78 para as 30 variáveis mais relevantes (e.x., *Destination Port*, *Packet Length Mean*, *Flow IAT*), descartando atributos redundantes de variância nula.

O desbalanceamento intrínseco de classes, mesmo após tentativa de balanceamento por agrupamento, foi tratado via *Synthetic Minority Over-sampling Technique* (SMOTE) (Chawla et al., 2002), gerando dados sintéticos por interpolação linear (*k-Nearest Neighbors* – k-NN). Para evitar o vazamento de dados (*data leakage*), adotou-se uma técnica onde a geração de dados sintéticos foi aplicada exclusivamente ao conjunto de Treino (70%), equalizando as classes de ataque à categoria Benigna (Figura 3), enquanto o conjunto de Teste (30%) preservou a distribuição original para simular fielmente um cenário de produção.

Tabela 1. Mapeamento da Taxonomia (Original → Agrupado).

Macro-Categoria	**Classes Originais (CIC-IDS2017)**
Benigno	BENIGN
DoS/DDoS	DoS Hulk, DoS GoldenEye, DoS slowloris, DoS Slowhttptest, DDoS
PortScan	PortScan
Brute Force	FTP-Patator, SSH-Patator
Web Attack	Web Attack-Brute Force, XSS, Sql Injection
Bot/Infiltration	Bot, Infiltration, Heartbleed

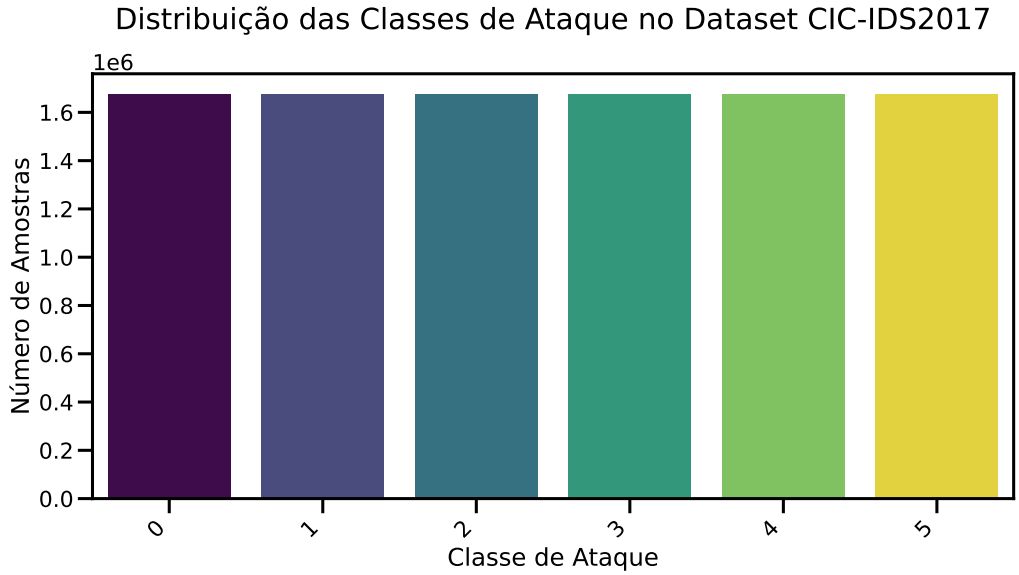


Figura 3. Distribuição das classes no conjunto de treino após aplicação do SMOTE.

3.3. Modelagem, Explicabilidade e Ambiente Experimental

A avaliação comparativa abrange seis algoritmos representativos de diferentes paradigmas de classificação. Como *baselines*, adotaram-se o Naive Bayes (GaussianNB), assumindo independência condicional entre atributos, e o Linear SVC (*Support Vector Classifier*), verificando a separabilidade linear das fronteiras. O núcleo da análise concentra-se em *ensembles* de árvores: o Random Forest (*Bagging*) (Breiman, 2001), selecionado pela robustez à variância; e algoritmos de *Gradient Boosting* sequenciais, especificamente o XGBoost (regularização avançada), LightGBM (Ke et al., 2017) (crescimento *leaf-wise* e binagem de histogramas para eficiência) e CatBoost (Prokhorenkova et al., 2018) (otimizado para categóricos e redução de viés via *Ordered Boosting*).

Para transcender a métrica de acurácia e auditar a lógica decisória, integrou-se o *framework SHapley Additive exPlanations* (SHAP) (Lundberg and Lee, 2017). Fundamentado na Teoria dos Jogos Cooperativos, o método foi priorizado em detrimento da *Feature Importance* tradicional por garantir consistência global (estabilidade de valores frente a alterações no modelo) e acurácia local, permitindo a auditoria individualizada de fluxos via *Force Plots* para detecção de correlações espúrias. A implementação utilizou o otimizador *TreeExplainer*, viabilizando o cálculo exato dos valores Shapley em tempo polinomial.

Todos os experimentos foram conduzidos em *hardware* de uso geral (AMD Ryzen 7 5700U 1.80 GHz, 12 GB RAM, Windows 11 64-bit), operando exclusivamente via CPU. Esta restrição de *hardware* visa demonstrar a viabilidade de implantação dos modelos propostos em cenários de *Edge Computing* com recursos limitados, sem dependência de aceleradores dedicados.

Vale ressaltar que os experimentos foram conduzidos em um ambiente com restrições de *hardware* e sistema operacional de propósito geral (Windows 11), operando exclusivamente via CPU. Em um cenário de produção real, a implementação desses modelos em sistemas operacionais otimizados para redes (e.x., Linux *Bare-metal*) ou a

utilização de aceleração de *hardware* dedicada (como GPUs ou FPGAs) tenderia a reduzir ainda mais a latência absoluta de inferência, superando os resultados conservadores aqui apresentados.

É importante ressaltar que o tempo de inferência reportado (e.x., 0,65 s para o CatBoost) refere-se ao processamento vetorial em lote (*batch processing*) de todo o conjunto de teste ($N = 504.160$ amostras). Esta métrica representa a capacidade máxima de vazão (*throughput*) do modelo em cenários de alto tráfego, e não a latência de ida e volta de um único pacote isolado.

4. Resultados e Discussão

Esta seção apresenta a avaliação experimental dos modelos analisados para a detecção de intrusões no *dataset* CIC-IDS2017. A análise dos resultados foi estruturada em três eixos fundamentais para validar a aplicabilidade do sistema em cenários reais: a eficácia preditiva, comparando métricas globais de classificação entre os seis algoritmos avaliados; a eficiência operacional, onde se discute o *trade-off* entre a acurácia obtida e o custo computacional de treinamento e inferência; e a interpretabilidade, utilizando técnicas de XAI para auditar as decisões do modelo e garantir que o aprendizado não foi enviesado por ruídos estatísticos. Todas as métricas reportadas a seguir referem-se exclusivamente ao conjunto de teste (30% dos dados originais), preservado durante a etapa de balanceamento, garantindo assim a isenção e a capacidade de generalização dos modelos frente a novos vetores de ataque.

A Tabela 2 consolida os resultados obtidos pelos seis classificadores avaliados no conjunto de teste. As métricas de Acurácia, Precisão, *Recall* e *F1-Score* foram calculadas utilizando a média ponderada (*weighted average*) para considerar o desbalanceamento residual das classes no conjunto de teste. O tempo de treinamento refere-se ao tempo total de ajuste do modelo sobre os dados de treino balanceados. A leitura dos dados revela um cenário de *trade-off* estratégico entre as abordagens. Observa-se que o Random Forest se estabelece como o "Campeão de Acurácia", obtendo o melhor desempenho bruto (99,64%), mas apresentando a maior latência de inferência entre os modelos de alta performance (2,22 s), devido à necessidade de percorrer centenas de árvores profundas para cada decisão. Em contrapartida, o LightGBM destacou-se na fase de ajuste, sendo seis vezes mais rápido que o Random Forest para treinar, o que o torna ideal para ambientes que exigem atualizações constantes do modelo. Já o CatBoost, embora possua o maior custo de treinamento (¿1h), demonstrou ser o mais eficiente para operação em tempo real, com um tempo de inferência de apenas 0,65 s, sendo 3,4 vezes mais rápido que o Random Forest na borda (*edge*). Esta descoberta reposiciona o CatBoost como o "Vice-Campeão Operacional": ele sacrifica o tempo de treinamento (que ocorre *offline*) para entregar a menor latência possível durante a detecção ativa, uma característica crítica para redes de alta velocidade (Gbps).

A Figura 4 fundamenta a seleção do classificador conforme o requisito do IDS: o Random Forest é indicado para cenários de auditoria forense que exigem precisão máxima (99,64%), enquanto CatBoost e LightGBM são preferíveis para detecção ativa em tempo real sob alta carga de tráfego. O desempenho superior do Random Forest (F1-Score de 99,68%) sobre modelos lineares valida o uso de estruturas hierárquicas para isolar anomalias no *dataset* CIC-IDS2017. Assim, os resultados estabelecem um guia prático

Tabela 2. Resumo das métricas de desempenho, tempo de treinamento e inferência (Dados de Teste).

Modelo	Acurácia	F1-Score	Precisão	Recall	Tempo Treino (s)	Tempo Inferência (s)
Random Forest	**0,9964**	**0,9968**	**0,9974**	**0,9964**	1437,95	2,22
LightGBM	0,9922	0,9942	0,9969	0,9922	**222,74**	2,71
XGBoost	0,9920	0,9941	0,9970	0,9920	351,70	1,42
CatBoost	0,9912	0,9933	0,9963	0,9912	5668,79	**0,65**
LinearSVC	0,7085	0,7942	0,9498	0,7085	3416,71	0,28
Naive Bayes	0,5727	0,6836	0,9160	0,5727	10,97	0,78

para implementações que priorizam ou a integridade da detecção via (Random Forest) ou o rendimento operacional (via modelos de *Boosting*).

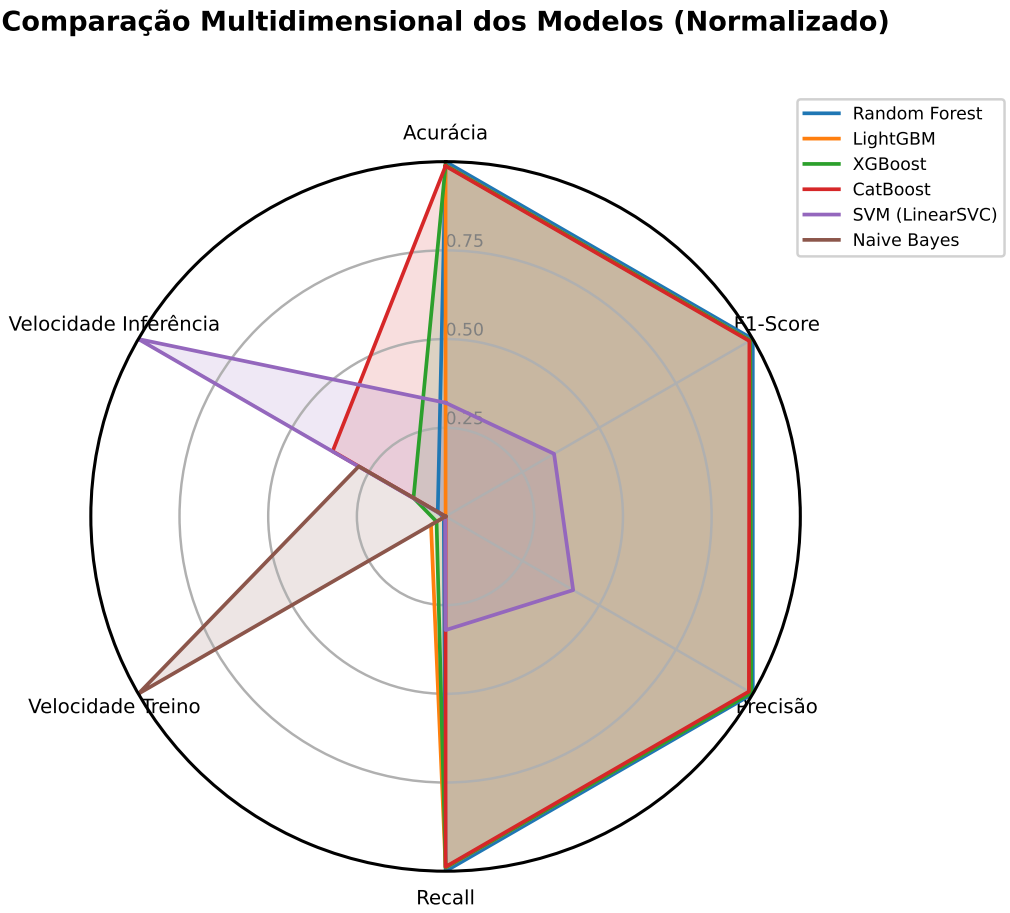


Figura 4. Comparativo multidimensional dos classificadores.

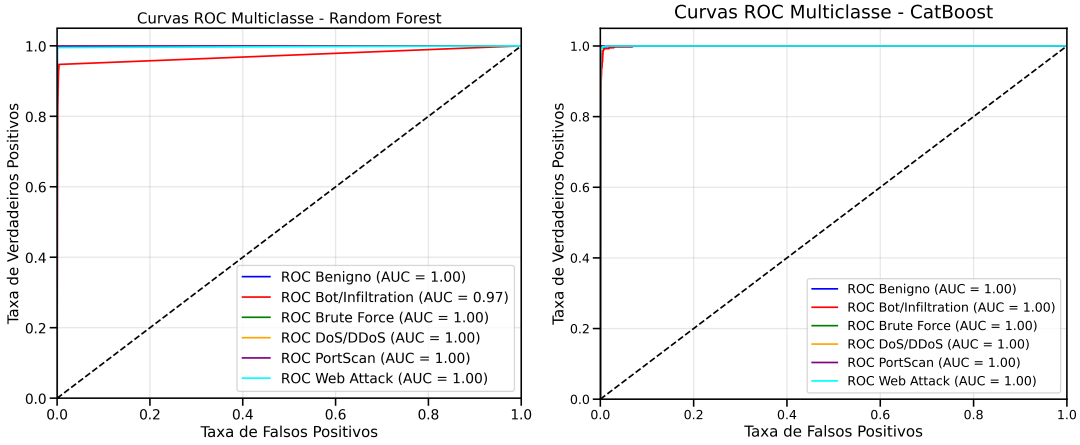
A robustez na classificação é detalhada na Matriz de Confusão (Tabela 3), que revela a capacidade do modelo em distinguir classes críticas com precisão quase perfeita. Destaca-se o desempenho na classe *Benigno*, onde o modelo minimizou drasticamente os falsos positivos, uma característica vital para evitar a interrupção de operações legítimas. Adicionalmente, ataques volumétricos como *DoS/DDoS* e *PortScan*, que possuem assinaturas estatísticas distintas (ex: alta frequência de pacotes em curto intervalo), foram classificados com precisão superior a 99%. A principal dificuldade residiu na distinção

entre ataques do tipo *Web Attack* e tráfego normal, devido à semelhança no padrão de transporte (HTTPS), mas ainda assim o modelo manteve uma taxa de detecção aceitável para uso em produção.

Tabela 3. Matriz de Confusão Normalizada do Random Forest (Dados de Teste).

Predito / Real	*Classe Predita*					
	Benigno	Bot/Infilt.	Brute Force	DoS/DDoS	PortScan	Web Attack
Benigno	**1.00**	0.00	0.00	0.00	0.00	0.00
Bot/Infiltration	0.22	**0.78**	0.00	0.00	0.00	0.00
Brute Force	0.01	0.00	**0.99**	0.00	0.00	0.00
DoS/DDoS	0.00	0.00	0.00	**1.00**	0.00	0.00
PortScan	0.01	0.00	0.00	0.00	**0.99**	0.00
Web Attack	0.01	0.00	0.00	0.01	0.00	**0.98**

Corroborando a estabilidade do classificador, a análise da Curva ROC Multiclasse (Figura 5) dos modelos Random Forest e CatBoost demonstra que a Área Sob a Curva (AUC) dos modelos atingiu valores próximos de 1.0 para todas as macro-categorias, em que o CatBoost foi superior apenas na classe de *Bot/Infiltration* onde obteve máxima de 1.0 sob os 0.97 do Random Forest . Isso indica que o Random Forest mantém altas taxas de Verdadeiros Positivos (*True Positive Rate* - TPR) mesmo quando exigido a operar com taxas de Falsos Positivos (*False Positive Rate* - FPR) extremamente baixas, provando que o modelo aprendeu fronteiras de decisão nítidas e não depende de ajustes finos de *threshold*. Além disso, diferentemente de modelos "caixa-preta" puros, o Random Forest oferece transparência através da métrica de Diminuição Média da Impureza (*Mean Decrease Impurity* - MDI). A análise dos atributos revelou que características como *Destination Port* e *Packet Length Mean* dominam a hierarquia de decisão. Este comportamento é tecnicamente coerente com a natureza dos ataques de rede, onde vetores como *Brute Force* visam portas específicas e DoS geram padrões repetitivos de tamanho de pacote, reforçando que o aprendizado baseou-se em padrões causais e não em ruídos estatísticos.



(a) Random Forest (b) CatBoost

Figura 5. Curvas ROC por classe para Random Forest e CatBoost (AUC $\approx$ 1.0).

Apesar da excelência preditiva do Random Forest, a avaliação do tempo de in-

ferência impõe uma redefinição da hierarquia de modelos para cenários de alta disponibilidade, revelando um gargalo de latência que pode inviabilizar sua aplicação em redes de altíssima velocidade (como *backbones* ou 5G). Os dados experimentais demonstram uma troca clara entre precisão e velocidade: ao optar pelo CatBoost, observa-se um decréscimo marginal de 0,52% na Acurácia global e 0,35% no *F1-Score*, perdas estatisticamente mínimas para a maioria das aplicações comerciais. Em contrapartida, o ganho de velocidade é expressivo, com o tempo de inferência caindo de 2,22 segundos para 0,65 segundos, representando uma aceleração de 3,4 vezes. Considerando a classificação de 504.160 fluxos, isso resulta em uma vazão de lote (*Batch Throughput*) expressiva de aproximadamente 775.630 fluxos por segundo. Tal capacidade, aliada à arquitetura de árvores simétricas (*Oblivious Trees*) do CatBoost, qualifica-o como a solução ideal para dispositivos de borda e *firewalls* em linha, enquanto o Random Forest permanece a escolha superior para auditoria forense.

A superioridade operacional do CatBoost observada neste cenário, embora contraste com *benchmarks* clássicos do LightGBM (Ke et al., 2017), pode ser atribuída à sua arquitetura de Árvores Simétricas (*Oblivious Trees*). Diferentemente da estratégia de crescimento *leaf-wise* do LightGBM, que pode gerar estruturas irregulares e dependentes de otimizações de histograma, as árvores simétricas do CatBoost permitem uma execução vetorial eficiente em CPUs de propósito geral. Adicionalmente, o tratamento nativo de variáveis categóricas pelo algoritmo preserva a densidade informacional de atributos de rede (como protocolos e flags), evitando a sobrecarga computacional frequentemente associada ao pré-processamento de dados esparsos.

Para avaliar a capacidade de generalização dos modelos e mitigar o risco de *overfitting*, empregou-se a técnica de validação cruzada estratificada com k=5 dobras. A acurácia média reportada para cada algoritmo foi obtida por meio da média aritmética dos resultados colhidos em cada uma das cinco iterações independentes, servindo como um indicador de robustez estatística frente a diferentes subconjuntos de dados.

Os resultados desse procedimento evidenciaram a clara superioridade dos métodos baseados em árvores de decisão e *ensemble*. A arquitetura Random Forest manteve a liderança em termos de estabilidade, atingindo uma acurácia média de 99,81% com um desvio padrão de apenas 0,0001. Esse elevado grau de consistência foi compartilhado pelos algoritmos de *Gradient Boosting*, com o XGBoost (99,76%) e o CatBoost (99,70%) apresentando variabilidade igualmente ínfima, indicando um comportamento preditivo altamente confiável através das diferentes dobras. O LightGBM, embora figure entre as soluções de alta performance, exibiu uma oscilação ligeiramente superior (0,0033), com média de 99,36%.

Em contrapartida, os modelos Linear SVC e Naive Bayes demonstraram menor robustez, com médias de 91,87% e 83,95%, respectivamente. Além da menor precisão, esses modelos exibiram maior sensibilidade às variações dos dados, com o Naive Bayes apresentando um desvio padrão de 0,0050. Essa disparidade reforça que a fronteira de decisão que separa o tráfego benigno dos ataques no *dataset* CIC-IDS2017 possui uma complexidade não linear, que é capturada com maior precisão por estruturas hierárquicas de particionamento do que por abordagens baseadas em hiperplanos simples ou na suposição de independência condicional de atributos.

Por fim, a adoção de modelos de *Black Box* exige a mitigação da desconfiança analítica através de técnicas de XAI. Para validar a "lógica interna" do classificador, aplicou-se o *framework* SHAP. A Figura 6 apresenta o gráfico de abelhas (*Beeswarm plot*), cuja análise visual permite extrair *insights* cruciais. Observa-se que valores baixos de *Destination Port* (pontos azuis) frequentemente induzem à decisão de ataque, consistente com o alvo em portas de serviço conhecidas (0-1023), enquanto valores altos de *Packet Length Mean* (pontos vermelhos) estão associados a ataques, validando a detecção de exfiltração ou DoS volumétricos.

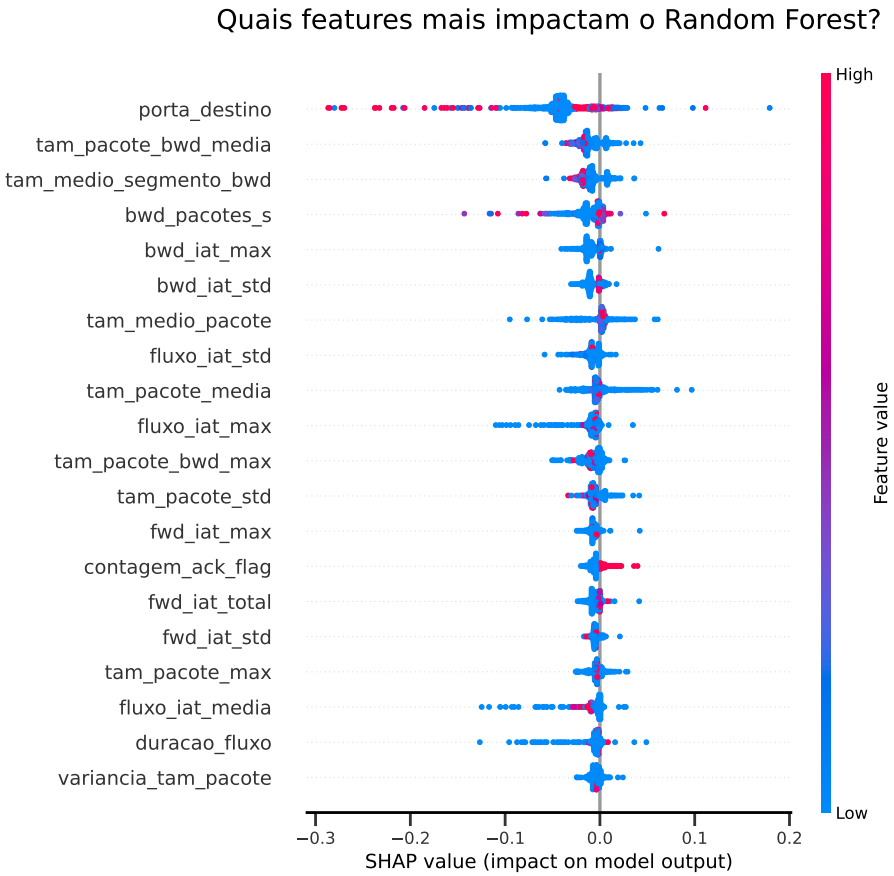


Figura 6. SHAP Summary Plot: Impacto global das características nas decisões do modelo.

Complementarmente, a análise local (Figura 7) decompõe vetorialmente uma predição de ataque, evidenciando como a combinação de fatores, como variância de *Flow IAT* e *flags* TCP, compõe o *output value*. Esta transparência revela que o aprendizado baseou-se em invariantes de protocolo e padrões causais, o que reforça a capacidade de generalização do sistema para redes além do CIC-IDS2017. O significado prático do XAI para a escolha do IDS reside em transformar o modelo de 'caixa-preta' em uma ferramenta auditável, garantindo que a detecção seja fundamentada em lógica de rede e não em ruídos estatísticos, o que reduz o risco de *overfitting* em cenários reais.

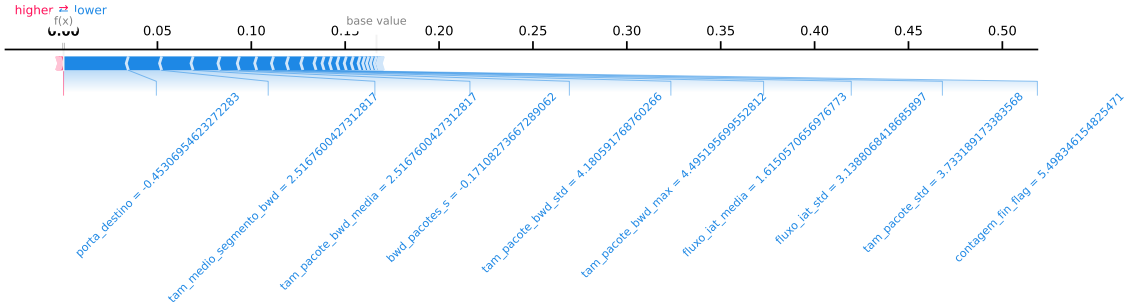


Figura 7. SHAP Force Plot: Decomposição vetorial de uma predição de ataque específica.

5. Conclusão e Trabalhos Futuros

A evolução dos vetores de ataque em redes de computadores modernas exige uma mudança de paradigma nos mecanismos de defesa, transicionando de assinaturas estáticas para modelos preditivos capazes de generalizar padrões anômalos. Este trabalho apresentou uma análise comparativa rigorosa de algoritmos de aprendizado de máquina aplicados ao *dataset* CIC-IDS2017, com foco no tripé: eficácia, eficiência e explicabilidade.

Os experimentos permitiram extrair três conclusões fundamentais. Primeiramente, algoritmos de *Ensemble* baseados em árvores superaram consistentemente abordagens lineares e probabilísticas. O Random Forest consolidou-se como o classificador mais robusto em métricas brutas, atingindo uma Acurácia de 99,65% e demonstrando resiliência na detecção de classes minoritárias e críticas, como *Web Attacks* e *Botnets*.

Em segundo lugar, a análise conjunta de desempenho e interpretabilidade fundamenta a proposição de uma Arquitetura Híbrida em Níveis (*Tiered Hybrid Architecture*). Os resultados posicionam o CatBoost como a solução definitiva para Defesa Ativa na Borda (*Active Edge Defense*), alavancando seu superior *Batch Throughput* para filtrar tráfego massivo em tempo real. Em contrapartida, o Random Forest assume o papel estratégico de Auditoria Forense e Explicabilidade em uma segunda linha de defesa. Esta segregação é vital pois permite que técnicas de XAI computacionalmente intensivas (como o cálculo de valores SHAP) sejam executadas seletivamente apenas sobre anomalias complexas, garantindo transparência diagnóstica para o analista de segurança sem comprometer a latência de processamento da rede na borda.

Por fim, a aplicação prática do *framework* SHAP validou a confiabilidade dessa camada forense, demonstrando que as decisões dos modelos baseiam-se em características causais de rede (e.x., portas de destino, *flags* TCP) e não em correlações espúrias. A capacidade de gerar explicações locais via *Force Plots* provou-se essencial para mitigar o problema da "caixa-preta", fornecendo o contexto necessário para a tomada de decisão humana em Centros de Operações de Segurança (SOCs).

Como trabalhos futuros, propõe-se: (i) a validação desta arquitetura híbrida em um ambiente de produção real, orquestrando a passagem de fluxo entre a borda (CatBoost) e o núcleo forense (Random Forest/SHAP); (ii) a avaliação da robustez dos modelos contra ataques adversariais (*Adversarial ML*), projetados para evadir a detecção estatística; e (iii) a extensão do estudo para o possível *dataset* mais recente, verificando a consistência dos resultados frente a novas topologias de ataque.

Disponibilidade de Artefatos

`https://github.com/jhonatasjgr/Deteccao_Intrusao_Redes`

Uso de Inteligência Artificial Generativa

Em conformidade com as diretrizes de conduta da Sociedade Brasileira de Computação (SBC), os autores declaram a utilização de ferramentas de Inteligência Artificial (IA) Generativa para auxiliar nas seguintes etapas deste trabalho:

- **Revisão Bibliográfica:** A ferramenta *NotebookLM* (Google) foi utilizada para indexar a base de referências e auxiliar na localização eficiente de trechos e conceitos específicos dentro dos artigos selecionados, otimizando o processo de consulta teórica.
- **Desenvolvimento e Formatação:** O modelo *Gemini* (Google) foi empregado como auxiliar na otimização de *scripts* em Python para geração de gráficos, na estruturação do código LaTeX e na revisão gramatical do texto.

Os autores ressaltam que a concepção da pesquisa, a seleção das fontes bibliográficas, a análise crítica dos resultados e a redação final do conteúdo são de autoria humana. As ferramentas de IA foram empregadas exclusivamente como instrumentos de apoio à produtividade, e os autores assumem integral responsabilidade pela originalidade, veracidade e integridade de todo o conteúdo aqui apresentado.

Referências

Anis, F. M., Alabdullatif, M., Aljbli, S., and Hammoudeh, M. (2025). A survey on the applications of deep learning in network intrusion detection systems to enhance network security. *IEEE Access*, 13:185348–185373.

Arreche, O., Guntur, T., and Abdallah, M. (2024). Xai-ids: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems. *Applied Sciences*, 14(10):4170.

Breiman, L. (2001). Random forests. *Machine Learning*, 45(1):5–32.

Chawla, N. V., Bowyer, K. W., Hall, L. O., and Kegelmeyer, W. P. (2002). Smote: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Elasaad, M. M. A., Sayed, S. G., and El-Dakroury, M. M. (2025). Aegisguard: A multi-stage hybrid intrusion detection system with optimized feature selection for industrial iot security. *Sensors*, 25(22):6958.

Grinsztajn, L., Oyallon, E., and Varoquaux, G. (2022). Why do tree-based models still outperform deep learning on tabular data? *Advances in Neural Information Processing Systems (NeurIPS)*, 35:507–520.

Hakami, H., Faheem, M., and Ahmad, M. B. (2025). Machine learning techniques for enhanced intrusion detection in iot security. *IEEE Access*, 13:31145–31158.

Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T.-Y. (2017). LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 3146–3154.

Khan, M. F., Hassan, M. M., Ferdous, S., Hussain, I., Akter, L., and Gupta, A. B. (2024a). Explainable ai and machine learning models for transparent and scalable intrusion detection systems. *Journal of Information Systems Engineering and Management*, 9(45).

Khan, N., Ahmad, K., Tamimi, A. A., Alani, M. M., Bermak, A., and Khalil, I. (2024b). Explainable ai-based intrusion detection system for industry 5.0: An overview of the literature, associated challenges, and potential research directions. *arXiv preprint arXiv:2408.03335*.

Lundberg, S. M. and Lee, S.-I. (2017). A unified approach to interpreting model predictions. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 4765–4774.

Nigar, N. and Mustafa, R. (2025). Enhanced intrusion detection via hybrid data resampling and feature optimization. *IEEE Access*, 13:149105–149120.

Prokhorenkova, L., Gusev, G., Vorobev, A., Dorogush, A. V., and Gulin, A. (2018). CatBoost: unbiased boosting with categorical features. In *Advances in Neural Information Processing Systems (NeurIPS)*, pages 6638–6648.

Santos, K. and Miani, R. (2025). Impacto da redução de dimensão e seleção de atributos na generalização de modelos de detecção de intrusão. In *Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 728–741, Porto Alegre, RS, Brasil. SBC.

Schmidt, T., Granville, L., and Schaeffer-Filho, A. (2025). Analyzing energy and performance trade-offs for network anomaly detection based on deep learning. In *Anais do XLIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 224–237, Porto Alegre, RS, Brasil. SBC.

Sharafaldin, I., Lashkari, A. H., and Ghorbani, A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, pages 108–116. SciTePress.

Siganos, M., Radoglou-Grammatikis, P., Kotsiuba, I., Markakis, E., and Moscholios, I. (2022). Explainable ai-based intrusion detection in the internet of things. *IEEE Transactions on Industrial Informatics*.

Wali, S., Farrukh, Y. A., Khan, I., and Bastian, N. D. (2025). Explainable ai and random forest based reliable intrusion detection system. *Computers & Security*, 157:104542.