Universidade Estadual de Londrina

MATEUS KOMARCHESQUI

INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL PARA OTIMIZAÇÃO DE SISTEMAS DE DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES

LONDRINA
2026

MATEUS KOMARCHESQUI

INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL PARA OTIMIZAÇÃO DE SISTEMAS DE DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

Orientador: Prof. Dr. Mario Lemes Proença Jr.

Coorientador: Prof. Dr. Luiz Fernando Carvalho

LONDRINA

2026

Ficha de identificação da obra elaborada pelo autor, através do Programa de Geração Automática do Sistema de Bibliotecas da UEL

K81i Komarchesqui, Mateus.
Inteligência Artificial Explicável para Otimização de Sistemas de Detecção de Anomalias em Redes de Computadores / Mateus Komarchesqui. - Londrina, 2026.
100 f. : il.

Orientador: Mario Lemes Proença Junior.
Coorientador: Luiz Fernando Carvalho.
Dissertação (Mestrado em Ciência da Computação) - Universidade Estadual de Londrina, Centro de Ciências Exatas, Programa de Pós-Graduação em Ciência da Computação, 2026.
Inclui bibliografia.

1. Explicações Aditivas de Shapley - Tese. 2. Otimização de Desempenho - Tese. 3. Sistema de Detecção de Intrusão - Tese. I. Lemes Proença Junior, Mario. II. Fernando Carvalho, Luiz. III. Universidade Estadual de Londrina. Centro de Ciências Exatas. Programa de Pós-Graduação em Ciência da Computação. IV. Título.

CDU 519

MATEUS KOMARCHESQUI

INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL PARA OTIMIZAÇÃO DE SISTEMAS DE DETECÇÃO DE ANOMALIAS EM REDES DE COMPUTADORES

Dissertação apresentada ao Programa de Mestrado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Mestre em Ciência da Computação.

BANCA EXAMINADORA

Orientador: Prof. Dr. Mario Lemes Proença Jr.
Universidade Estadual de Londrina

Prof. Dr. Wesley Attrot
Universidade Estadual de Londrina

Prof.ª Dr.ª Maria Bernadete de Morais França
Universidade Estadual de Londrina

Londrina, 25 de agosto de 2026.

AGRADECIMENTOS

Agradeço a todas as pessoas e instituições que contribuíram, de forma direta ou indireta, para a realização desta pesquisa.

Em primeiro lugar, à Fundação Araucária de Apoio ao Desenvolvimento Científico e Tecnológico do Estado do Paraná (FA) e ao Programa de Pós-Graduação em Ciência da Computação da Universidade Estadual de Londrina (UEL), pelo apoio institucional e pela concessão da bolsa de estudos, elementos fundamentais para viabilizar minha dedicação exclusiva a este projeto.

Ao meu orientador, Professor Doutor Mario Lemes Proença Jr., expresso minha profunda gratidão. Seus ensinamentos em gerência de redes de computadores, formalismo acadêmico, resiliência e disciplina, aliados ao seu constante apoio e confiança em meu trabalho, foram guias indispensáveis nesta jornada. Ao meu coorientador, Professor Doutor Luiz Fernando Carvalho, pela disponibilidade, orientação e compartilhamento de sua experiência.

Aos demais professores e servidores do Departamento de Computação e da UEL, que contribuíram para minha formação e mantêm o *campus* como um ambiente de excelência profissional e propício à produção de conhecimento.

Aos meus colegas e amigos de laboratório e pesquisa, agradeço pelas discussões enriquecedoras, pela dedicação mútua e pelo companheirismo diário que tornaram este percurso mais leve e produtivo.

Por fim, dedico um agradecimento especial à minha família e à minha namorada, pelo amor incondicional, pelo incentivo constante e pela compreensão diante das minhas ausências.

KOMARCHESQUI, M.. **Inteligência Artificial Explicável para Otimização de Sistemas de Detecção de Anomalias em Redes de Computadores**. 2026. 100f. Dissertação (Mestrado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2026.

RESUMO

Ataques cibernéticos visam degradar a qualidade de serviços de uma rede, motivados por interesses que variam de políticos a monetários. Sistemas de detecção de intrusão, frequentemente implementados com algoritmos de aprendizado profundo, modelam o comportamento do usuário legítimo ajustando múltiplos parâmetros numéricos durante o treinamento. Como esse processo não é transparente para agentes humanos, estratégias de inteligência artificial explicável emergem como métodos práticos para elucidar o funcionamento interno desses sistemas. A maioria dos autores utiliza técnicas de explicabilidade, como *SHapley Additive exPlanations* (SHAP), apenas para exibir a magnitude da importância das características, desconsiderando a orientação dos impactos, a distinção entre classes de tráfego e a dimensão temporal do tráfego. Essa análise superficial oculta comportamentos enviesados: características prejudiciais permanecem no modelo e sustentam altas taxas de falsos alarmes e de ataques indetectados, penalizando usuários legítimos e sobrecarregando a triagem de incidentes. Esta dissertação propõe uma metodologia cíclica e transparente para explicar e otimizar sistemas de detecção de intrusão caixa-preta por meio de uma análise profunda de explicações de SHAP, aproveitando seu potencial para alcançar tanto um treinamento quanto um produto final explicáveis. Propõe-se um algoritmo de subamostragem não supervisionado, baseado em agrupamento hierárquico por densidade, que viabiliza computacionalmente o KernelSHAP sem comprometer a representatividade dos dados. As explicações estáticas e temporais são avaliadas de forma isolada por categoria de tráfego, considerando magnitude e orientação das contribuições, decaimento de influência e efeito de atraso temporal, o que fundamenta o descarte transparente das características enviesadas e o retreino do sistema. Realizam-se experimentos com dois sistemas de arquiteturas e paradigmas distintos: um baseado em redes adversárias generativas, proposto por autor externo, e outro em autocodificadores variacionais β. Para o primeiro, o coeficiente de correlação de Matthews (MCC) elevou-se de 0,804 para 0,851 no conjunto CIC-DDoS2019 e de 0,703 para 0,916 no CSE-CIC-IDS2018, com reduções de 26,6% e 92,3% nos falsos negativos. O segundo obteve avanços absolutos de 0,116 e 0,154 no MCC para essas mesmas bases e elevou o MCC médio de 0,928 para 0,952 no Orion2026, mitigando a penalização de tráfego legítimo entre diferentes cardinalidades de atacantes e vítimas. Conclui-se que a otimização guiada por SHAP é agnóstica à arquitetura e ao cenário de dados, entregando um sistema mais assertivo, mais enxuto e auditável, capaz de reduzir a saturação operacional da gerência de redes.

Palavras-chave: Explicações Aditivas de Shapley. Otimização de Desempenho. Redes Adversárias Generativas. Autocodificadores Variacionais. Sistema de Detecção de Intrusão.

KOMARCHESQUI, M.. **eXplainable Artificial Intelligence for Optimization of Anomaly Detection Systems in Computer Networks**. 2026. 100p. Master's Thesis (Master in Science in Computer Science) – State University of Londrina, Londrina, 2026.

ABSTRACT

Cyberattacks aim to degrade the quality of network services, motivated by interests ranging from political to monetary. Intrusion detection systems, frequently implemented with deep learning algorithms, model legitimate user behavior by adjusting multiple numerical parameters during training. Since this process is not transparent to human agents, explainable artificial intelligence strategies emerge as practical methods to elucidate the inner workings of these systems. Most authors employ explainability techniques, such as *SHapley Additive exPlanations* (SHAP), merely to display the magnitude of feature importance, disregarding the orientation of the impacts, the distinction between traffic classes, and the temporal dimension of network traffic. This superficial analysis conceals biased behaviors: harmful features remain in the model and sustain high rates of false alarms and undetected attacks, penalizing legitimate users and overloading incident triage. This dissertation proposes a cyclic and transparent methodology to explain and optimize black-box intrusion detection systems through a deep analysis of SHAP explanations, leveraging their potential to achieve both an explainable training and final product. An unsupervised undersampling algorithm, based on hierarchical density-based clustering, is proposed to make KernelSHAP computationally feasible without compromising data representativeness. Static and temporal explanations are assessed separately by traffic category, considering the magnitude and orientation of the contributions, influence decay, and time lag effects, which grounds the transparent discarding of biased features and the retraining of the system. Experiments are conducted with two systems of distinct architectures and learning paradigms: one based on generative adversarial networks, proposed by an external author, and another on β-variational autoencoders. For the former, the Matthews correlation coefficient (MCC) rose from 0.804 to 0.851 on the CIC-DDoS2019 dataset and from 0.703 to 0.916 on CSE-CIC-IDS2018, with reductions of 26.6% and 92.3% in false negatives. The latter achieved absolute gains of 0.116 and 0.154 in MCC for these same datasets and raised the average MCC from 0.928 to 0.952 on Orion2026, mitigating the penalty on legitimate traffic across different attacker-victim cardinalities. It is concluded that SHAP-guided optimization is agnostic to the architecture and to the data scenario, delivering a system that is more assertive, leaner, and auditable, capable of reducing the operational saturation of network management.

Keywords: SHapley Additive exPlanations. Performance Optimization. Generative Adversarial Networks. Variational Autoencoders. Intrusion Detection System.

LISTA DE ILUSTRAÇÕES

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

1D-CNN	*One-dimensional Convolutional Neural Network*
AIOps	Inteligência Artificial para Operações de TI
ALE	*Accumulated Local Effects*
BYOD	*Bring Your Own Device*
β-VAE	β-*Variational Autoencoder*
CAM	*Class Activation Map*
CAV	*Concept Activation Vectors*
CNN	*Convolutional Neural Network*
DDoS	*Distributed Denial of Service*
DeepLIFT	*Deep Learning Important FeaTures*
DL	*Deep Learning*
DNS	*Domain Name System*
DoH	*DNS over HTTPS*
DoS	*Denial of Service*
DTD	*Deep Taylor Decomposition*
EDA	*Exploratory Data Analysis*
ELBO	*Evidence Lower Bound*
EVT	*Extreme Value Theory*
FLOPs	*Floating-Point Operations*
FN	Falso Negativo
FP	Falso Positivo
GAN	*Generative Adversarial Network*
Grad-CAM	*Gradient-weighted Class Activation Mapping*
HDBSCAN	*Hierarchical Density-Based Spatial Clustering of Applications with Noise*

HTTPS	*HyperText Transfer Protocol Secure*
IA	Inteligência Artificial
ICE	*Individual Conditional Expectations*
ICMP	*Internet Control Message Protocol*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
IP	*Internet Protocol*
KDE	*Kernel Density Estimation*
LDAP	*Lightweight Directory Access Protocol*
LIME	*Local Interpretable Model-agnostic Explanations*
LRP	*Layer-wise Relevance Propagation*
LSTM	*Long Short-Term Memory*
MCC	*Matthews Correlation Coefficient*
NIDS	*Network Intrusion Detection System*
OOM	*Out of Memory*
PDP	*Partial Dependence Plots*
PFI	*Permutation Feature Importance*
p.p.	Ponto Percentual
RAM	*Random Access Memory*
RISE	*Randomized Input Sampling to Provide Explanations*
SD-WAN	*Software-Defined Wide Area Network*
SHAP	*SHapley Additive exPlanations*
SMOTE	*Synthetic Minority Oversampling Technique*
TCN	*Temporal Convolutional Network*
TCP	*Transmission Control Protocol*
TFN	Taxa de Falso Negativo

TFP	Taxa de Falso Positivo
TI	Tecnologia da Informação
UDP	*User Datagram Protocol*
VAE	*Variational Autoencoder*
VN	Verdadeiro Negativo
VP	Verdadeiro Positivo
XAI	*eXplainable Artificial Intelligence*

LISTA DE SÍMBOLOS

α	Taxa de amostragem
β	Termo de penalidade/regularização do Autocodificador Variacional
Γ	Classe de ataque no conjunto de dados
γ	*Cluster* de dados
$\Delta_i(S)$	Contribuição marginal da i-ésima característica
ϵ	Ruído aleatório amostrado de uma distribuição normal padrão
η	Parâmetros da rede neural Decodificadora
θ	Parâmetros da rede neural Geradora (p_θ ou G_θ)
λ	Parâmetro de suavização (largura de banda) do *kernel* Gaussiano
μ	Vetor de médias da distribuição latente
$\pi_{\tilde{x}}(b)$	Peso de *kernel* associado à instância perturbada
ρ	Parâmetros da rede neural Codificadora
σ	Vetor de desvios padrão da distribuição latente
τ	Duração da janela deslizante de tráfego (em segundos)
ϕ	Parâmetros da rede neural Discriminadora
ψ_0	Predição base / saída média do modelo
ψ_i	Valor de Shapley da i-ésima característica
$\Omega(g)$	Termo de regularização de complexidade do modelo
$\odot$	Multiplicação elemento a elemento (Produto de Hadamard)
$*$	Operação de convolução
b	Vetor binário indicativo da presença de características
c, l	Dimensões do filtro matricial (*kernel*) da convolução
D	Conjunto de dados tabular original
D'	Conjunto de dados subamostrado

D_{KL}	Divergência de Kullback-Leibler
D_ϕ	Função probabilística do Discriminador
$\mathbb{E}[\cdot]$	Valor esperado (esperança matemática)
F	Conjunto total de características (*features*)
$f(x)$	Previsão/saída do modelo caixa-preta original
$\mathcal{G}$	Conjunto de modelos lineares de aproximação
G_θ	Função de mapeamento do Gerador
$g(b)$	Modelo linear de aproximação local (substituição)
$h_x(b)$	Função de mapeamento do espaço simplificado binário para os dados originais
$\mathcal{I}$	Imagem (matriz de características) de entrada da rede convolucional
$\mathcal{I}'$	Imagem resultante (mapa de características) da convolução
$\mathcal{J}$	Função de perda do otimizador KernelSHAP
k	Filtro matricial (*kernel*) da rede neural convolucional
$\mathcal{L}_D$	Função de perda do Discriminador
$\mathcal{L}_G$	Função de perda do Gerador
$\mathcal{L}_{\beta\text{-VAE}}$	Função de perda do Autocodificador Variacional β
M	Número de componentes interpretáveis / dimensionalidade das características
$\mathcal{N}(0, I)$	Distribuição normal padrão
n	Número de instâncias de dados no *cluster*
p_G	Distribuição probabilística induzida pelo Gerador
p_r	Distribuição probabilística dos dados reais observados
$p_\eta(x\|z)$	Rede Decodificadora baseada na variável latente
$q_\rho(z\|x)$	Rede Codificadora (aproximação da distribuição posterior)
S	Coalizão (subconjunto) de características presentes
s^*	Tamanho de *cluster* subótimo determinado via Otimização Bayesiana

u, v	Coordenadas de uma posição específica na matriz convolucional
x	Instância de dados originais / amostra do espaço de dados
$\tilde{x}$	Versão perturbada / simplificada da entrada original
x'	Amostra de dados reconstruída pelo decodificador
y	Rótulo verdadeiro da amostra classificatória
z	Variável ou vetor amostrado no espaço latente

SUMÁRIO

1 INTRODUÇÃO

A telecomunicação moderna é cada vez mais definida pela convergência de tecnologias como o 5G, a Internet das Coisas (do inglês *Internet of Things*, IoT) e as Redes de Longa Distância Definidas por *Software* (*Software-Defined Wide Area Network*, SD-WAN) [1, 2, 3, 4]. A integração dessas infraestruturas de rede é impulsionada pela consolidação dos ambientes de trabalho híbridos, caracterizados por uma força móvel e distribuída, além de políticas como Traga Seu Próprio Dispositivo (*Bring Your Own Device*, BYOD) [5]. Diante desse cenário dinâmico e descentralizado, a gestão de redes deve garantir, de forma contínua e integrada, a prestação de serviços, ao mesmo tempo em que atende aos requisitos de falha, configuração, contabilização, desempenho e segurança [6].

À medida que as fronteiras das redes e a interconectividade dos dispositivos se expandem, o volume e a velocidade do tráfego ampliam drasticamente a superfície de ataque [5, 7]. Agentes maliciosos exploram essa complexidade com ameaças cada vez mais sofisticadas [8], desde a varredura de vulnerabilidades até a injeção de códigos ou interrupção de serviços [9, 10]. Nesse contexto, os métodos tradicionais de monitoramento baseados em regras tornaram-se impraticáveis devido à sua dificuldade de escalonamento [11, 12].

A literatura aponta a transição para a Inteligência Artificial para Operações de TI (AIOps) como uma solução plausível para esse gargalo [13, 14, 15, 16, 17, 18]. Destacam-se os Sistemas de Detecção de Intrusão (*Intrusion Detection Systems*, IDS) impulsionados por Aprendizado Profundo (*Deep Learning*, DL), que monitoram de forma autônoma a telemetria de fluxo contínuo. Ao aprender padrões emaranhados em grandes volumes de dados [19, 20, 21], esses modelos identificam comportamentos espúrios que destoam dos usuários legítimos [22, 23]. Uma vez detectada a anomalia, o sistema pode gerar alertas para triagem manual ou acionar algoritmos de reforço em Sistemas de Prevenção de Intrusão para o ajuste dinâmico de políticas [24, 25, 26].

Contudo, além de estarem sujeitos a altas taxas de falsos alarmes e ataques não detectados, os modelos de DL operam ajustando iterativamente múltiplos pesos por tentativa e erro, o que os caracteriza como "caixas-pretas". Depender de soluções críticas de segurança cuja lógica interna é inacessível torna-se perigoso, seja pela presença camuflada de vieses ou pela simples falta de transparência. Para mitigar esse problema e aumentar a confiança no processo de decisão da Inteligência Artificial (IA), a adoção de métodos de Inteligência Artificial Explicável (*eXplainable Artificial Intelligence*, XAI) emerge como uma etapa analítica adicional indispensável no desenvolvimento de soluções de Aprendizado Profundo [27].

A seleção de um algoritmo de XAI depende de fatores como o escopo da explicação, a fase de treino e o tipo de dados que se deseja explicar. Para modelos previamente treinados com dados tabulares, como é o caso dos IDS caixa-preta, uma das estratégias mais consolidadas é o *SHapley Additive exPlanations* (SHAP), devido à sua robustez matemática baseada na teoria dos jogos e à consistência nas explicações globais [28, 29]. O SHAP determina a contribuição de cada característica dos dados usando os valores de Shapley [30], oriundos da teoria dos jogos cooperativos, oferecendo explicações locais (ocorrências individuais) e globais (tendências gerais) por meio de agrupamento de elucidações singulares. Explicações globais são comumente apresentadas em gráficos de enxame (*beeswarm plots*), nos quais as múltiplas instâncias individuais são espalhadas ao longo de um eixo, ilustrando as influências positivas e negativas de cada característica no processo decisório do modelo [31, 32, 33]. No contexto da classificação binária de tráfego, um valor SHAP positivo comumente indica maior probabilidade de uma amostra ser espúria.

Apesar de sua utilidade, a análise do SHAP na literatura recente frequentemente se restringe à apresentação do valor médio absoluto, o qual reflete a magnitude das contribuições, mas falha em indicar a orientação positiva ou negativa desses impactos [34, 35, 36, 37]. Tanto a análise superficial de gráficos de enxame quanto o uso exclusivo de médias absolutas podem ocultar comportamentos indesejados e vieses do modelo.

Problemas adicionais advêm da dependência de subconjuntos de dados não representativos, da agregação indiscriminada de categorias de tráfego distintas e da avaliação estática das explicações. A desconsideração da dimensão temporal do tráfego de rede compromete a análise de sua natureza contínua e dinâmica, aspecto essencial para refletir o desenvolvimento dos ataques cibernéticos ao longo de janelas temporais. A relevância de determinadas características pode não permanecer constante, revelando importâncias divergentes entre as fases de início, pico e término dos eventos espúrios. Negligenciar esses aspectos culmina em uma interpretação equivocada e incompleta do processo de decisão do modelo [31, 33, 35].

Para suprir essas lacunas, esta dissertação propõe um método transparente para a otimização de IDS por meio da análise de explicações de SHAP. Como estudo de caso, dois IDS baseados em aprendizado profundo são considerados em três cenários de dados públicos: CIC-DDoS2019 [38], CSE-CIC-IDS2018 [39] e Orion2026 [40]. Esses conjuntos abrangem cardinalidades distintas entre atacantes e vítimas, além de tipos de ataque e padrões de tráfego benigno diversos, o que permite avaliar a generalização e a robustez do método proposto em diferentes contextos de rede. Este método busca endereçar três limitações observadas nos trabalhos do estado da arte:

- Primeiro, para combater o problema de subconjuntos de dados não representativos, realiza-se uma Análise Exploratória de Dados (*Exploratory Data Analysis*, EDA) e introduz-se um novo método de subamostragem baseado em agrupamento hierár-

quico não supervisionado, cuja robustez é validada em comparação com um método estratificado aleatório usando a distância de Wasserstein.

- Segundo, em vez de agregar categorias de tráfego e focar apenas na magnitude do impacto, analisam-se os gráficos de enxame de SHAP de forma granular por categoria, avaliando a magnitude e a orientação da contribuição de cada característica. Essa abordagem permite identificar quando características de impacto de alta magnitude prejudicam o desempenho do modelo.
- Terceiro, estende-se a análise para a dimensão temporal, verificando o decaimento da influência e os efeitos de defasagem temporal. Esses aspectos, frequentemente ignorados em XAI estático, avaliam, respectivamente, o enfraquecimento da influência das características ao longo do tempo e o efeito de valores passados de características de séries temporais nas previsões atuais.

A metodologia deste trabalho é dividida em cinco etapas: Sistema de Detecção de Intrusão, Módulo de Explicação, Análise por Classes de Tráfego, Seleção de Atributos e Treinamento do Sistema. O fluxo metodológico é iniciado pela submissão de um IDS já treinado ao Módulo de Explicação. Neste, os dados são subamostrados e injetados no algoritmo de XAI, produzindo explicações sobre o processo decisório do sistema. As explicações são analisadas de forma granular por categoria de dados. Caso nenhum viés seja detectado, o fluxo chega ao fim, produzindo um sistema explicável. Caso contrário, uma Seleção de Atributos é realizada, dando início ao Treinamento do Sistema, que produz uma nova instância de IDS.

Essa metodologia cíclica é repetida até que não sejam identificadas mais características enviesadas eliminávéis. O resultado é um processo transparente de seleção de características que descarta variáveis prejudiciais, culminando em um IDS otimizado com desempenho aprimorado, maior equidade (*fairness*, mitigando previsões tendenciosas) e explicabilidade, o que minimiza erros classificatórios e a carga de trabalho operacional. Diante do cenário exposto, as principais contribuições deste trabalho são:

- A proposição de uma estratégia de subamostragem de dados não supervisionada baseada em agrupamento (*clustering*) para a extração de subconjuntos representativos. Esse conjunto de amostras alimenta o algoritmo SHAP, balanceando o compromisso (*trade-off*) entre o uso de recursos computacionais e a fidedignidade das explicações.
- A investigação das explicações globais de SHAP de maneira isolada por categoria de tráfego de rede, permitindo que o comportamento de cada tipo de ataque ou tráfego legítimo seja analisado sem a interferência dos demais.
- A análise detalhada dessas explicações considerando as influências positivas e negativas de cada característica nas predições do modelo.

- O desenvolvimento de um processo de seleção de características transparente e diretamente relacionado às importâncias destacadas pelo SHAP.
- A elucidação do processo de tomada de decisão do sistema otimizado final, visando reforçar sua confiabilidade.
- A avaliação da metodologia proposta em conjuntos de dados distintos, contemplando diferentes cardinalidades entre atacantes e vítimas, tipos de ataque e padrões de tráfego benigno. Além disso, dois sistemas caixa-preta de arquiteturas e paradigmas de aprendizado distintos são avaliados. Essa diversidade evidencia a capacidade de generalização do método frente a cenários heterogêneos de rede e de IDS.

O restante deste trabalho está organizado em cinco capítulos. O Capítulo 2 apresenta a fundamentação teórica base para o entendimento dos conceitos utilizados neste trabalho. O Capítulo 3 compreende uma revisão bibliográfica de trabalhos do estado da arte que empregam algoritmos de XAI tanto para explicação dos sistemas quanto para otimizações de desempenho de detecção. O Capítulo 4 detalha a metodologia de otimização guiada por XAI proposta neste trabalho, enquanto o Capítulo 5 expõe os experimentos conduzidos e os resultados obtidos. O Capítulo 6 conclui o trabalho.

2 FUNDAMENTAÇÃO TEÓRICA

2.1 Detecção de Intrusão em Redes

Ataques e intrusões exploram vulnerabilidades de *hardware*, *software* ou humanas de forma estratégica a fim de comprometer a confidencialidade, disponibilidade e integridade dos dados que trafegam e são armazenados na rede de computadores. Por tirarem proveito de brechas de segurança, o padrão de tráfego desses eventos espúrios tende a divergir do comportamento normal de um usuário legítimo, podendo ser abstraído como uma anomalia de rede [41].

A detecção dessas ocorrências é uma tarefa central dos Sistemas de Detecção de Intrusão de Redes (*Network Intrusion Detection System*, NIDS), que monitoram continuamente o tráfego com o objetivo de identificar atividades espúrias. Os NIDS são implementados com base em duas abordagens principais: aquelas baseadas em anomalia e as baseadas em assinatura [42, 43]. Os sistemas de assinatura armazenam padrões de tráfego de diversos tipos de ameaças de segurança e detectam ocorrências similares a padrões conhecidos, emitindo um alerta ao administrador da rede. Essa abordagem tende a gerar menos alertas falsos; contudo, carece da capacidade de detectar ataques desconhecidos (*zero-day*). Sistemas baseados em anomalia modelam o tráfego legítimo, considerando toda ocorrência divergente como anômala e, portanto, uma ameaça [44, 45, 46]. Essa estratégia é mais sensível a variações do tráfego e propensa a gerar falsos alarmes; entretanto, apresenta maior potencial para detectar ataques *zero-day* em virtude de sua capacidade inerente de modelagem. Pela maior versatilidade em detecção, sistemas baseados em anomalia são os mais empregados em trabalhos do estado da arte.

A modelagem de padrões de tráfego não é uma tarefa trivial ou tratável por meio de regras explícitas, uma vez que o comportamento legítimo da rede pode variar de acordo com os serviços acessados, usuários ativos, além de mudar ao longo do tempo. Técnicas de aprendizado de máquina buscam aprender padrões inerentes dos dados a partir do processo de treinamento, sem a necessidade de intervenção humana. Por esse motivo, NIDS propostos por trabalhos do estado da arte frequentemente empregam aprendizado de máquina [18, 43].

Técnicas de aprendizado de máquina tradicionais, como máquinas de vetor de suporte [47], árvores de decisão [48] e algoritmos de agrupamento [49], são limitadas quanto ao volume de dados e à complexidade dos padrões que conseguem modelar. Com o aumento da velocidade e diversidade do tráfego nas redes de computadores modernas, essas abordagens enfrentam dificuldade de escalabilidade e generalização. A fim de abordar essas limitações, arquiteturas de redes neurais profundas e especializadas têm se destacado na implementação de Sistemas de Detecção de Intrusão [41, 50].

2.2 Rede Adversária Generativa

A Rede Adversária Generativa (*Generative Adversarial Network*, GAN) é uma arquitetura de rede neural profunda proposta por Goodfellow *et al.* [51] que visa aprender a sintetizar amostras a partir de um conjunto de dados reais.

O conjunto de dados reais pode ser representado como uma distribuição probabilística p_r [52]. Essa distribuição é desconhecida, portanto, procura-se aproximá-la por meio de um modelo genérico p_θ, parametrizado por redes neurais. Para que p_θ represente de fato uma distribuição de probabilidade, é necessário que satisfaça dois critérios fundamentais: não negatividade e normalização [53, 54]. Matematicamente, para uma instância de dados x, deve valer $p_\theta(x) \geq 0$ e $\int_x p_\theta(x)dx = 1$. Contudo, em espaços de alta dimensão a integral de normalização é intratável, o que inviabiliza o cálculo direto de $p_\theta(x)$. A solução é introduzir uma variável latente z, com distribuição conhecida e normalizada $p_z(z)$, de modo que a distribuição no espaço dos dados seja definida indiretamente pela transformação de z [55]. Assim, em vez de modelar $p_\theta(x)$ de forma explícita, o problema passa a ser o de aprender uma função que mapeie amostras do espaço latente para o espaço de dados.

No contexto das GANs, como ilustrado na Figura 1, essa função de mapeamento é implementada pelo Gerador G_θ. Dado um vetor latente $z \sim p_z$, o gerador produz uma amostra sintética $x = G_\theta(z)$. Essa operação induz uma distribuição p_G sobre o espaço de dados, que corresponde à concretização de p_θ. O objetivo do treino adversário é ajustar os parâmetros de G_θ de forma que a distribuição induzida p_G se aproxime da distribuição real p_r, tornando as amostras sintéticas indistinguíveis das reais para o Discriminador [56].

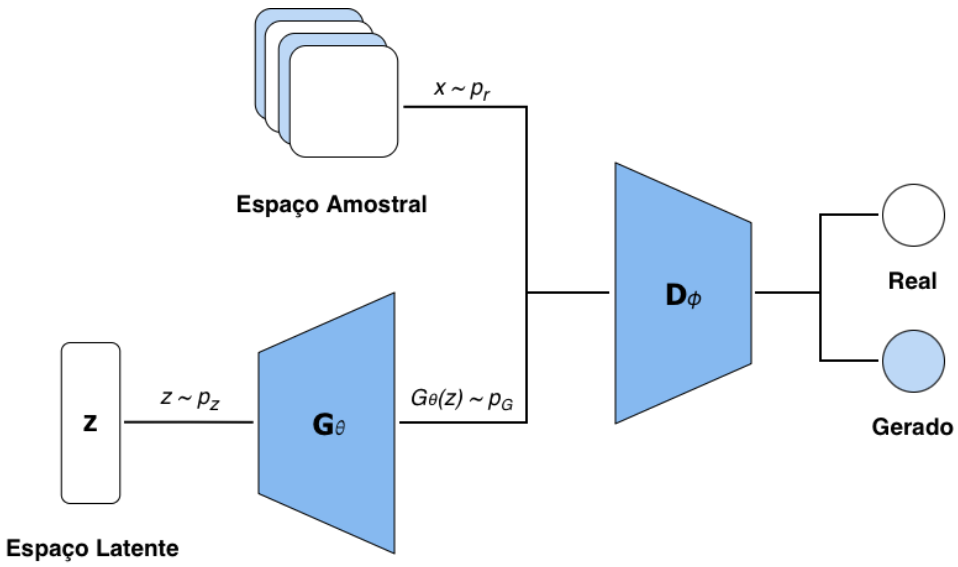

Figura 1 – Rede Adversária Generativa. **Fonte:** elaboração do próprio autor.

O Discriminador, denotado por D_ϕ, tem como objetivo distinguir entre amostras reais, extraídas de p_r, e amostras sintéticas, geradas a partir de p_G. Formalmente, o discriminador é uma função parametrizada que recebe uma instância x como entrada e retorna um valor escalar $D_\phi(x) \in [0; 1]$, interpretado como a probabilidade de x ter sido extraída da distribuição real. Tanto a saída do discriminador quanto a distribuição real podem ser consideradas variáveis de Bernoulli, em que o rótulo $y = 1$ representa uma amostra real ($x \sim p_r$) e $y = 0$ representa uma amostra sintética ($x \sim p_G$). A função de perda do discriminador busca alinhar sua predição $D_\phi(x)$ ao rótulo verdadeiro y [51], sendo expressa como a entropia cruzada binária na Equação seguinte,

$$\mathcal{L}_D(\phi \mid \theta) = -\mathbb{E}_{x\sim p_r}\Big[\log D_\phi(x)\Big] - \mathbb{E}_{z\sim p_z}\Big[\log\Big(1 - D_\phi(G_\theta(z))\Big)\Big]. \tag{2.1}$$

Visto que o treino da GAN é formulado como um jogo entre duas redes neurais, o Discriminador busca maximizar sua capacidade de distinguir entre amostras reais e sintéticas, enquanto o Gerador busca enganar o Discriminador ao produzir amostras que pareçam reais. Dessa forma, a função de perda do Gerador pode ser definida como o negativo da perda do Discriminador aplicada às amostras sintéticas, expressa em

$$\mathcal{L}_G(\theta \mid \phi) = \mathbb{E}_{z\sim p_z}\Big[\log\Big(1 - D_\phi(G_\theta(z))\Big)\Big]. \tag{2.2}$$

Enquanto D_ϕ tenta maximizar a probabilidade de classificar corretamente os dados, G_θ tenta minimizar essa mesma capacidade. Esse treino adversário pode ser modelado como um problema de otimização do tipo minimax, dado por

$$\min_\theta \max_\phi \; \mathbb{E}_{x\sim p_r}\Big[\log D_\phi(x)\Big] + \mathbb{E}_{z\sim p_z}\Big[\log\Big(1 - D_\phi(G_\theta(z))\Big)\Big]. \tag{2.3}$$

O treinamento deve finalizar quando o Equilíbrio de Nash se estabelece, um estado no qual tanto o Gerador quanto o Discriminador não conseguem diminuir suas funções de perda ao modificar seus parâmetros, assumindo que o outro modelo permanece estável.

Variações da GAN foram propostas devido a falhas do modelo tradicional, como o colapso de modo, a instabilidade de treino e a dissipação de gradientes [57, 58]. O colapso de modo ocorre quando o modelo perde capacidade de diversificar os exemplos sintetizados pelo gerador. A instabilidade no treino refere-se à dificuldade da GAN em alcançar o Equilíbrio de Nash, dado o caráter competitivo e altamente sensível da otimização conjunta entre gerador e discriminador. A dissipação de gradientes surge quando o discriminador se torna excessivamente eficaz em distinguir amostras reais de sintéticas, fazendo com que o gerador receba gradientes pequenos, impossibilitando a atualização suficiente de seus parâmetros para atingir desempenho comparável ao do discriminador. Dentre as variações que buscam solucionar esses problemas estão a GAN convolucional [59], a GAN condicional [60] e a GAN de Wasserstein [61].

2.3 Rede Neural Convolucional

A Rede Neural Convolucional (do inglês *Convolutional Neural Network*, CNN) é uma arquitetura de rede neural que visa extrair características de imagens a partir de filtragem. A primeira proposta bem-sucedida da CNN foi feita por LeCun *et al.* [62] em 1998 para reconhecer dígitos escritos à mão [63]. Para realizar essa tarefa, a CNN combina conceitos de convolução e agrupamento (*pooling*) com redes neurais densas.

A convolução pode ser definida como a transformação de uma imagem $\mathcal{I}$ de entrada, codificada como uma matriz de pixels, em uma imagem $\mathcal{I}'$ de saída. Essa transformação é baseada em um filtro matricial k, conhecido como *kernel*, com $(2c+1)$ colunas e $(2l+1)$ linhas, permitindo que a convolução seja centralizada na origem. Cada pixel de $\mathcal{I}'$, definido por uma coordenada (u, v) [53], decorre da função expressa em

$$\mathcal{I}'(u,v) = k * \mathcal{I} = \sum_{i=-c}^{c} \sum_{j=-l}^{l} k(i,j) \cdot \mathcal{I}(u-i, v-j). \tag{2.4}$$

A imagem $\mathcal{I}$ é varrida seguindo um passo, número de células que o *kernel* percorre na convolução, o que define parcialmente a presença de redução de dimensionalidade da imagem de entrada [53, 64]. O passo é combinado com o preenchimento, que busca encaixar o filtro nas bordas da imagem. Comumente o passo é definido com o valor um, gerando uma varredura pixel a pixel, e o preenchimento é utilizado para não gerar redução de dimensionalidade [52]. O processo de convolução busca modificar a imagem de entrada a fim de realçar ou extrair características da forma desejada [65]. Além disso, uma CNN pode empregar múltiplos filtros durante uma convolução, o que culmina na geração de uma matriz resultante para cada varredura individual. Esse conjunto de matrizes geradas por cada passagem do filtro é denominado mapa de características. Extrações mais complexas dependem de *kernels* mais elaborados e são afetadas pelo tamanho dos objetos alvo na imagem [66].

Para identificar objetos de diversos tamanhos, as Redes Neurais Convolucionais utilizam uma estratégia de subamostragem por região da imagem de entrada, processo conhecido como agrupamento, ou *pooling* [65]. O objetivo dessa operação é reduzir progressivamente a resolução espacial dos mapas de características. Essa diminuição confere menor carga computacional ao modelo. Além disso, o agrupamento opera deslizando uma janela sobre o mapa de características, aplicando uma função de agregação aos valores da janela, o que resulta em um valor para cada região.

Como retratado na Figura 2, a arquitetura da CNN comumente alterna camadas de convolução e *pooling*, construindo uma representação hierárquica da imagem de entrada. Os mapas de características resultantes dessas camadas ficam organizados em um tensor tridimensional (altura × largura × número de canais). A fim de integrar as carac-

terísticas extraídas com a rede neural, uma camada de achatamento (*flatten*) é aplicada [67]. Essa camada transforma o tensor multidimensional em um vetor unidimensional, que é entrada de uma rede neural composta por camadas densas e uma camada de saída. Geralmente a saída dessa rede é ativada pela função *softmax*, que atribui à imagem de entrada probabilidades de pertencer a cada uma das classes de saída [68].

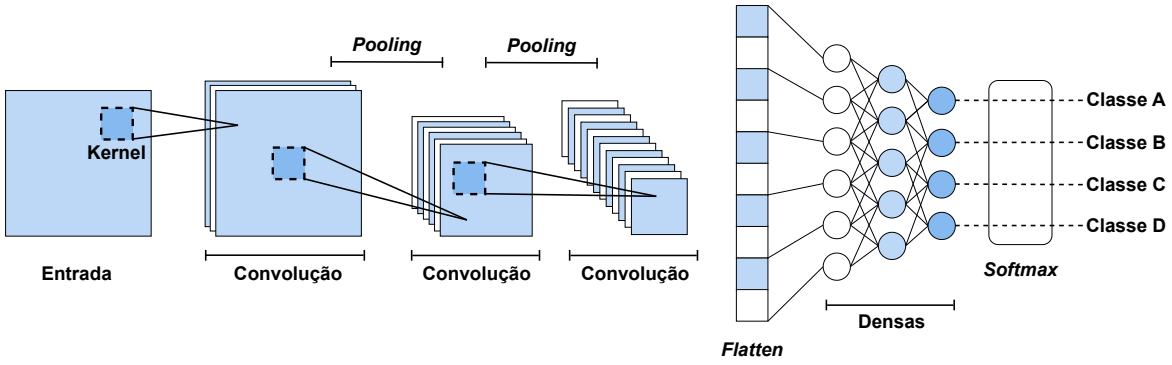

Figura 2 – Rede Neural Convolucional. **Fonte:** elaboração do próprio autor.

As Redes Neurais Convolucionais unidimensionais constituem uma variação da CNN bidimensional. A principal diferença reside na convolução e nos dados, pois, em vez de empregar um filtro bidimensional sobre dados matriciais, elas utilizam um vetor unidimensional em dados unidimensionais [67]. Essa implementação é aplicada neste trabalho deslizando o *kernel* sobre o eixo sequencial das estatísticas de tráfego de rede de computadores.

2.4 Autocodificador Variacional β

O Autocodificador Variacional (do inglês *Variational Autoencoder*, VAE), proposto por Kingma e Welling [69], é uma arquitetura de rede neural generativa fundamentada em inferência variacional bayesiana. Embora o VAE seja capaz de aprender representações latentes contínuas, suas variáveis frequentemente sofrem de emaranhamento, resultando em um espaço latente pouco interpretável no qual uma única dimensão pode codificar múltiplas características dos dados simultaneamente. Para contornar esse problema, o β-VAE foi introduzido por Higgins *et al.* [70] com o objetivo de induzir o desemaranhamento dessas representações. Essa abordagem obtém um espaço latente fatorado, no qual cada dimensão captura, de forma isolada, um único fator de variação estatisticamente independente presente nos dados originais.

Assim como nas redes GAN, assume-se que os dados observados x são gerados por um processo aleatório contínuo envolvendo uma variável latente não observada z. O β-VAE baseia-se em duas redes neurais parametrizadas conjuntas: o Codificador, denotado

por $q_\rho(z|x)$, que mapeia probabilisticamente os dados reais para uma distribuição no espaço latente; e o Decodificador, denotado por $p_\eta(x|z)$, que reconstrói os dados a partir das representações latentes amostradas [69].

A arquitetura que materializa esse processo de inferência e reconstrução é detalhada na Figura 3. O fluxo de dados se inicia com a inserção da amostra original x no codificador. Diferentemente de autocodificadores clássicos, a rede $q_\rho(z|x)$ não produz um único vetor latente, mas sim os parâmetros que definem uma distribuição de probabilidade para o dado inserido: um vetor de médias (μ) e um vetor de desvios padrão (σ). A partir desses parâmetros, realiza-se a amostragem do vetor latente z. Para que esse processo estocástico de amostragem permita o fluxo contínuo de gradientes durante o treinamento, aplica-se o truque da reparametrização (*reparameterization trick*). Por meio dessa técnica, z é calculado de forma determinística como $z = \mu + \sigma \odot \epsilon$, sendo ϵ um ruído aleatório amostrado de uma distribuição normal padrão $\mathcal{N}(0, \mathbf{I})$. O vetor amostrado z, que contém um resumo comprimido da entrada, é então repassado ao decodificador $p_\eta(x|z)$. A função final dessa rede é gerar a entrada reconstruída x' o mais fidedigna possível à amostra original ($x \approx x'$), denotada pela integração $p_\eta(x) = \int p_\eta(x|z)p(z)dz$.

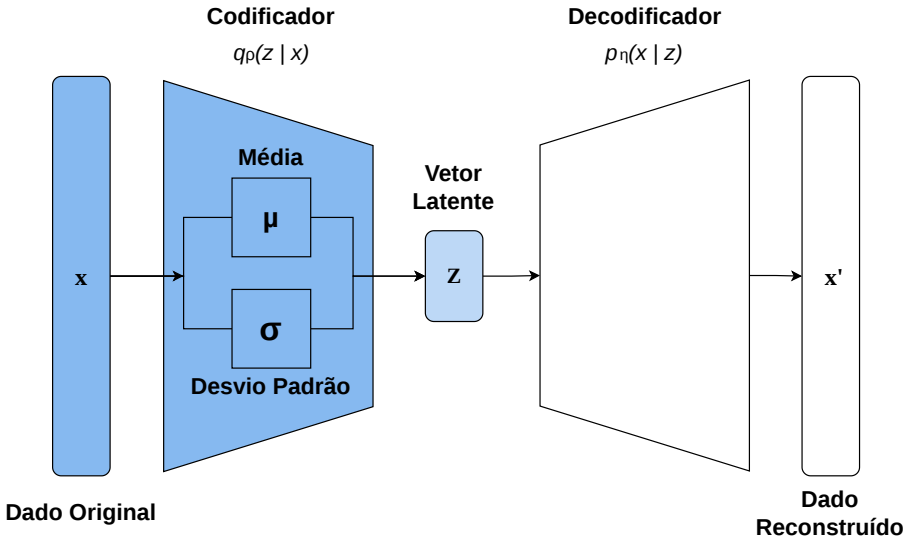

Figura 3 – Autocodificador Variacional. **Fonte:** elaboração do próprio autor.

Essa integração exige a marginalização sobre todo o espaço latente z. Como z opera em um espaço contínuo tipicamente multidimensional, o cálculo da integral e a determinação da verdadeira distribuição posterior $p_\eta(z|x)$ tornam-se computacionalmente intratáveis; portanto, para viabilizar o treinamento, o modelo emprega a inferência variacional bayesiana. O codificador $q_\rho(z|x)$ passa a atuar como uma aproximação para a distribuição posterior verdadeira, permitindo que os parâmetros sejam atualizados por meio da maximização do Limite Inferior das Evidências (*Evidence Lower Bound*, ELBO).

Dessa maneira, a função de perda do β-VAE é otimizada durante o treinamento e composta intrinsecamente por duas porções distintas, que perseguem objetivos concorrentes para a rede neural, representada por

$$\mathcal{L}_{\beta\text{-VAE}}(x, \eta, \rho) = -\mathbb{E}_{z\sim q_\rho(z|x)}\Big[\log p_\eta(x \mid z)\Big] + \beta \times D_{KL}\Big(q_\rho(z \mid x) \parallel p(z)\Big). \tag{2.5}$$

A primeira parcela corresponde ao termo de reconstrução do ELBO ($-\mathbb{E}_{z\sim q_\rho(z|x)}\Big[\log p_\eta(x \mid z)\Big]$) e diz respeito ao erro de reconstrução, cuja finalidade é avaliar a fidelidade e a precisão com que o decodificador consegue reproduzir os dados originais. A segunda parcela ($\beta \times D_{KL}$) atua como um termo de regularização probabilística, encarregado de estruturar e impor restrições à distribuição do espaço latente. É na inclusão do termo de penalidade β na segunda porção que o β-VAE se diferencia de seu predecessor, transformando o problema de otimização padrão em um cenário com restrições.

Quando $\beta = 1$, a função de perda se iguala à do VAE padrão. No entanto, ao definir $\beta > 1$, o modelo aplica uma penalidade maior sobre a divergência, criando um gargalo de informação no espaço latente. Sob essas condições, o codificador encontra suas capacidades de representação restringidas e não consegue mais manter codificações redundantes ou correlacionadas. Para minimizar a perda, o modelo é forçado a encontrar uma representação mais eficiente, isolando as características essenciais e estatisticamente independentes em dimensões distintas do vetor latente z. Essa penalidade introduz o desemaranhamento do espaço latente z.

Contudo, conforme o termo β é elevado, introduz-se um *trade-off*. Ao amplificar o peso do regularizador, a rede passa a priorizar a independência estrutural do espaço latente em detrimento do termo de reconstrução. O codificador, estrangulado pelo gargalo de informação, passa a descartar variações mais sutis dos dados originais para satisfazer a restrição matemática. O resultado empírico desse desequilíbrio é a geração de decodificações imprecisas ou incompletas. Portanto, a eficácia do β-VAE em reconstruir dados depende do balanceamento desse *trade-off* na calibração do hiperparâmetro β.

2.5 Inteligência Artificial Explicável

A Inteligência Artificial Explicável é o campo de estudo complementar à Inteligência Artificial, englobando técnicas de elucidação de modelos de IA caixas-pretas, cujo funcionamento interno não é inteligível ao ser humano. Os métodos de XAI empregam diferentes estratégias para tornar mais compreensível o processo de decisão ou a arquitetura das caixas-pretas, promovendo maior transparência, auditabilidade e confiança em soluções baseadas em IA [71].

Algoritmos de XAI classificados como métodos de explicação buscam detalhar o processo decisório dos modelos de IA, demonstrando as razões de predições ou comportamentos específicos; já aqueles voltados a racionalizar a estrutura são considerados métodos de interpretação. As técnicas de XAI também se organizam conforme o escopo de explicação fornecido e a fase de treinamento em que são aplicadas [72].

O conceito de escopo é cabível apenas a técnicas de explicação e é dividido em local e global. Explicações locais explicitam as intuições sobre o processo de classificação de instâncias individuais, enquanto explicações globais mostram tendências e noções gerais do modelo. Por focar em amostras individuais, o escopo local, na maioria das vezes, não é suficiente para elucidar o comportamento de um modelo de IA como um todo [73].

Ao longo do desenvolvimento do modelo, as técnicas de XAI podem ser aplicadas em três diferentes momentos: construção dos dados (*ante-hoc* ou *pre-hoc*), arquitetura do modelo (intrínseco ao modelo) e após o treino (*post-hoc*) [72, 13]. As estratégias *ante-hoc* buscam explicar os dados de treino, explicitando estatísticas de correlação, distribuições marginais e condicionais, presença de ocorrências atípicas (*outliers*), desequilíbrio entre classes, além de identificar atributos redundantes ou irrelevantes. Essa etapa engloba a Análise Exploratória de Dados, Engenharia de Características, Grafos de Conhecimento e Métodos de Sumarização de Conjuntos de Dados. Métodos intrínsecos ao modelo buscam investigar o processo de treinamento, criando modelos naturalmente mais compreensíveis. Essa categoria de XAI limita as famílias de modelos de IA a algoritmos interpretáveis, como regressões lineares, árvores de decisão, modelos baseados em regras e, em alguns casos, modelos lineares generalizados. Além disso, esses métodos exploram alterações na arquitetura de modelos caixas-pretas, como a hibridização de suas estruturas com algoritmos interpretáveis e a introdução de camadas de regularização. As abordagens XAI *post-hoc* são as mais utilizadas em trabalhos do estado da arte e abrangem métodos que explicam modelos já totalmente treinados.

A relação entre XAI e IA convencional quanto ao desempenho preditivo depende da categoria de técnica empregada. As abordagens *post-hoc* não modificam o modelo explicado nem suas saídas, limitando-se a acrescentar uma justificativa às predições já produzidas, de modo que não representam ganho de acurácia, mas de transparência e auditabilidade. Os métodos intrínsecos, por sua vez, restringem a modelagem a algoritmos interpretáveis, o que pode implicar perda de desempenho em problemas complexos e caracteriza um compromisso entre acurácia e interpretabilidade [74].

A Figura 4 ilustra a organização dos métodos de XAI *post-hoc* em seis categorias principais, seguindo a taxonomia proposta por Ali *et al.* [75]: atribuição, visualização, baseados em exemplos, teoria dos jogos, extração de conhecimento e métodos neurais.

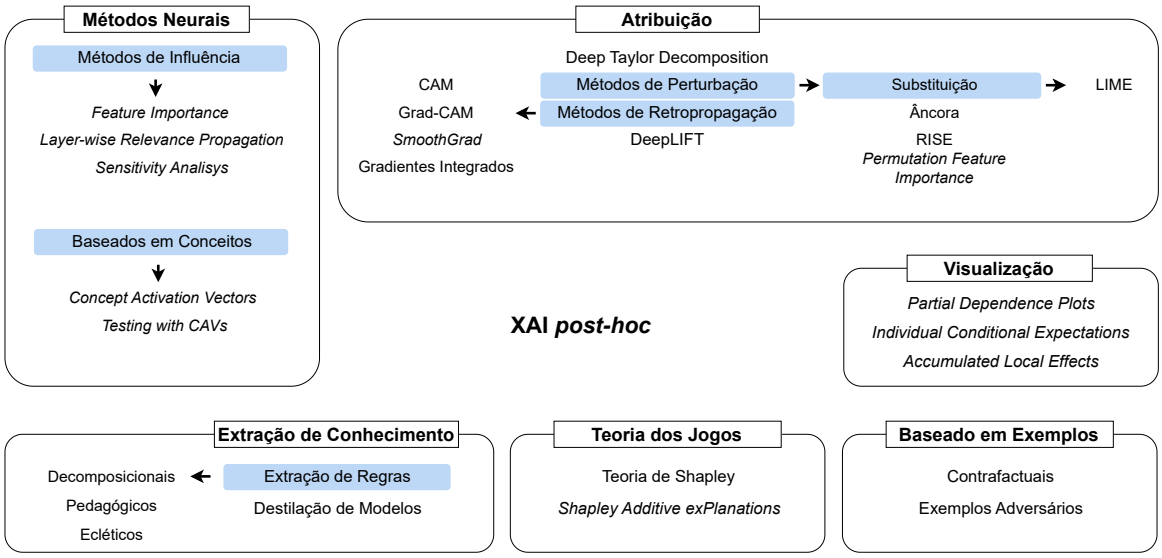

Figura 4 – Categorias de XAI *post-hoc*. **Fonte:** elaboração do próprio autor.

Ferramentas *post-hoc* de atribuição visam identificar quais parcelas de uma entrada de dados contribuíram para uma predição específica. Essas podem ser categorizadas em diferentes famílias:

- *Deep Taylor Decomposition* (DTD): consiste em somar pontuações de relevância de características para elucidar o comportamento local do modelo [76];
- Métodos de Perturbação: calculam a atribuição de cada característica dos dados modificando, deletando ou mascarando a instância de entrada e observando o impacto que essas mudanças acarretam na saída do modelo. Ferramentas de perturbação incluem estratégias de substituição de modelo, como o *Local Interpretable Model-agnostic Explanations* (LIME), que ajusta um modelo interpretável sobre as amostras localmente perturbadas; Âncora, que busca uma regra de decisão que explica uma predição individual; o *Randomized Input Sampling to Provide Explanations* (RISE), voltado para visão computacional, que embaralha as regiões de uma imagem para estimar a relevância de cada parte na classificação final; e o *Permutation Feature Importance* (PFI), que mede o quanto a permutação de uma variável compromete o desempenho do modelo [77, 78];
- Retropropagação: determina os valores de atribuição para as características com base em retropropagação em uma Rede Neural. Por exemplo, o *Class Activation Map* (CAM) identifica áreas discriminativas de uma imagem que contribuem para sua classificação, enquanto o *Gradient-weighted Class Activation Mapping* (Grad-CAM) evolui o CAM para utilizar informações de gradiente. Mapas de Saliência calculam o gradiente da função de perda em relação aos pixels de uma imagem

de entrada para realçar as áreas de maior impacto preditivo. *SmoothGrad* reduz o ruído dos mapas de saliência, adicionando ruído Gaussiano à imagem de entrada e calculando a média dos mapas de sensibilidade. Finalmente, Gradientes Integrados acumulam gradientes ao longo da retropropagação levando em conta uma linha base, como uma imagem preta, a fim de suprimir ruídos e fornecer atribuições mais confiáveis [79];

- *Deep Learning Important FeaTures* (DeepLIFT): atribui pontuação de importância às variáveis de entrada com base em seus desvios em relação a um ponto de referência pré-definido, geralmente uma imagem neutra definida pelo usuário. A qualidade e a consistência das explicações podem ser sensíveis à escolha da condição de referência [80].

Ferramentas *post-hoc* de visualização concentram-se em investigar padrões subjacentes em modelos de IA a partir da visualização de representações comportamentais. Essa categoria de XAI compreende os algoritmos:

- *Partial Dependence Plots* (PDP): exibe uma previsão média do modelo caixa-preta quando uma das características dos dados é alterada ao longo de sua gama de valores, do valor mínimo ao máximo, enquanto as demais são fixadas. Essa técnica assume independência entre as variáveis [81];
- *Individual Conditional Expectations* (ICE): mostra para cada instância de dados o impacto na previsão do modelo ao alterar o valor de uma única característica enquanto as demais permanecem nos valores originais da amostra. Assim como o PDP, o ICE assume que as características não possuem correlação [82];
- *Accumulated Local Effects* (ALE): estima o impacto de uma variável sobre a predição do modelo calculando variações locais médias ao longo de intervalos do seu domínio, sem assumir independência entre as características [83].

As abordagens *post-hoc* baseadas em exemplos geram previsões para uma entrada fornecida e as comparam com amostras de treino usando métricas de distância. Dois principais algoritmos compõem essa família XAI:

- Contrafactuais: apresentam qual a mínima mudança em uma instância levaria o modelo caixa-preta a gerar uma predição alternativa [84];
- Exemplos Adversários: identificam perturbações, geralmente imperceptíveis ao agente humano, nos dados de entrada que induzem o modelo caixa-preta a mudar sua predição. Essas perturbações revelam sensibilidade do modelo e podem ser usadas para explicar seu processo decisório por meio de contrastes entre instâncias similares [85].

Os métodos de XAI baseados em teoria dos jogos empregam heurísticas da área de jogos cooperativos para explicar modelos de IA. Essa categoria compreende o algoritmo mais utilizado para explicações *post-hoc* de dados tabulares em trabalhos do estado da arte, o *SHapley Additive exPlanations* (SHAP). O SHAP ressignifica a teoria de Shapley, que visa calcular a contribuição de cada jogador em um jogo cooperativo, para determinar qual a importância de cada variável dos dados na previsão de uma instância. Essa ferramenta combina de maneira aditiva os impactos de cada característica em múltiplas amostras de dados, culminando em uma explicação global [30].

A extração de conhecimento refere-se a abordagens que objetivam recuperar o conhecimento aprendido por um modelo caixa-preta. Essa categoria possui duas ferramentas principais:

- Extração de Regras: aproxima regras de "SE-ENTÃO", que estabelecem que uma condição leva a uma determinada predição, e regras "M-de-N", utilizando expressões booleanas para determinar se um número mínimo de condições é satisfeito dentro de um conjunto total de restrições. Os métodos de extração se dividem em três tipos: os decomposicionais, que analisam cada neurônio internamente; os pedagógicos, que observam a entrada e saída da rede; e os ecléticos, que combinam os dois para gerar explicações mais completas [86, 87];
- Destilação de Modelos: treina um modelo mais simples e interpretável (aluno) para imitar o comportamento de um modelo caixa-preta (professor). O aluno aprende tanto as respostas quanto as intuições do professor [88].

Métodos neurais visam explicar redes neurais analisando predições individuais, simplificando modelos ou visualizando padrões aprendidos. Essa abordagem divide-se em duas principais estratégias:

- Métodos de Influência: medem o quanto cada entrada afeta a saída da rede. Essa abordagem inclui ferramentas como Importância de Características (*Feature Importance*), que avalia a importância de variáveis ao medir o impacto negativo que têm no desempenho do modelo; *Layer-wise Relevance Propagation* (LRP), que distribui a relevância da predição das saídas para as entradas, buscando determinar quais partes da entrada foram mais impactantes na predição levando em conta os pesos dos neurônios; e Análise de Sensibilidade (*Sensitivity Analysis*), que perturba entradas para observar a estabilidade da saída [74];
- Métodos Baseados em Conceitos: buscam traduzir os padrões internos do modelo caixa-preta em conceitos acessíveis ao agente humano. Essa técnica compreende o *Concept Activation Vectors* (CAV), representação vetorial de conceitos obtidos a partir da ativação de uma camada quando submetida a exemplos que possuem ou

não o conceito que se deseja explorar; e o *Testing with CAVs*, indicação da influência de cada conceito na predição do modelo caixa-preta a partir da aproximação de uma entrada e um conceito [89].

Quanto ao domínio de aplicação, a XAI é cabível aos mesmos contextos da IA convencional, embora nem todo modelo seja explicável na mesma medida. Em arquiteturas de grande escala, como os Grandes Modelos de Linguagem, não há uma única técnica capaz de elucidar o sistema integralmente, mas sim ferramentas que esclarecem parcelas de seu funcionamento, como a inspeção de mapas de atenção, a atribuição de importância a *tokens* de entrada e a análise de representações internas. Ainda assim, o valor explicativo desses mecanismos é objeto de disputa: Jain e Wallace [90] mostram que os pesos de atenção frequentemente não se correlacionam com medidas de importância baseadas em gradiente, enquanto Wiegreffe e Pinter [91] contestam essa conclusão, argumentando que ela depende da definição de explicação adotada. A explicação obtida nesses modelos é, portanto, necessariamente parcial, o que reforça o caráter da XAI como ferramenta de apoio à análise e não como garantia de compreensão completa do modelo.

Independentemente da categoria, a explicabilidade implica custo computacional adicional. Métodos de perturbação, como LIME e SHAP, exigem inferências suplementares para cada instância explicada, e o cálculo exato dos valores de Shapley cresce exponencialmente com o número de atributos [30]. Métodos contrafactuais, por sua vez, resolvem um problema de otimização a cada explicação gerada [84]. Esse sobrecusto, aliado ao fato de nem toda aplicação demandar justificativa para suas decisões, explica por que a XAI não é empregada de forma indiscriminada, concentrando-se em domínios críticos ou regulados, como saúde, concessão de crédito e segurança cibernética, além de auditorias e da depuração de modelos durante o desenvolvimento.

No contexto de detecção de intrusão em redes de computadores, técnicas de XAI que buscam explicar o conjunto de dados e a importância das variáveis do modelo completamente treinado são as mais empregadas, especialmente a Análise Exploratória de Dados, SHAP e LIME [13, 75]. Este trabalho estuda a aplicação de Análise Exploratória de Dados como um ponto de partida antes do treino de modelos caixas-pretas a fim de determinar a distribuição e correlação dos dados, auxiliando em uma análise mais minuciosa tanto do conjunto de dados quanto do sistema de detecção de intrusão.

Este trabalho emprega, ainda, uma variação menos computacionalmente custosa do SHAP, o KernelSHAP, após o treino do modelo para explicar seu processo decisório. Embora técnicas como LIME, ICE, LRP, DeepLIFT, ALE, PDP e PFI ofereçam explicações baseadas em atributos, suas abordagens não atendem plenamente às necessidades deste trabalho. LIME e ICE se apoiam em perturbações locais nas instâncias para estimar a relevância de atributos, enquanto LRP e DeepLIFT propagam pontuações de contribuição de uma amostra camada a camada na rede. Essas abordagens tendem a

produzir explicações inconsistentes entre instâncias, o que impossibilita a análise de padrões globais. ALE e PDP calculam efeitos marginais médios, desconsiderando interações complexas entre variáveis. Finalmente, o PFI busca medir a importância de cada atributo pela perda de desempenho ao permutá-lo, mas ignora interações entre variáveis e não gera uma decomposição direta do comportamento do modelo, sendo também sensível a correlações nos dados. O emprego de SHAP aumenta a transparência do sistema, agregando confiabilidade à solução de segurança, além de possibilitar a identificação de vieses que influenciam de forma negativa as predições do modelo.

2.5.1 LIME

Local Interpretable Model-agnostic Explanations (LIME) é uma técnica que explica a predição de qualquer classificador de forma local, aproximando o modelo complexo por um modelo linear mais simples na vizinhança da predição a ser explicada [92].

O LIME utiliza um modelo linear $g(b)$ para aproximar localmente a predição do modelo original $f(x)$, em que b é um vetor binário ($b \in \{0,1\}^M$), e M é o número de componentes interpretáveis (no contexto de dados tabulares, o número de características). Esse vetor binário indica a presença da i-ésima característica (b_i é 1 se presente, 0 caso contrário) em uma versão perturbada da entrada simplificada $\tilde{x}$. Quando uma característica está ausente, ela é substituída por um valor de referência, como o valor esperado $\mathbb{E}[x_i]$. A função de mapeamento h_x, responsável por mapear b para a entrada original x, a qual $\odot$ representa a multiplicação elemento a elemento, é expressa em

$$h_x(b) = x \odot b + \mathbb{E}[x] \odot (1 - b). \tag{2.6}$$

O modelo local do LIME, $g(b)$, segue uma atribuição de característica aditiva, conforme

$$g(b) = \psi_0 + \sum_{i=1}^{M} \psi_i b_i, \tag{2.7}$$

em que $\psi_0 = \mathbb{E}[f(x)]$ é a predição base, ou seja, a saída média do modelo.

Sendo $\mathcal{G}$ o conjunto de modelos lineares, o modelo g é obtido minimizando a função de perda heurística denotada em

$$\min_{g \in \mathcal{G}} \sum_{b} \pi_{\tilde{x}}(b)[f(h_x(b)) - g(b)]^2 + \Omega(g). \tag{2.8}$$

O termo $\pi_{\tilde{x}}(b)$ é um peso de *kernel* que atribui importâncias diferentes para cada instância de b; já $\Omega(g)$ é um termo de regularização para evitar sobreajuste.

No entanto, a escolha arbitrária do *kernel* de ponderação no LIME pode criar aproximações que divergem do modelo original, enquanto o termo de regularização pode suprimir a verdadeira importância das características.

2.5.2 SHapley Additive exPlanations

O *SHapley Additive exPlanations* é uma técnica concebida por Lundberg e Lee em 2017 [30], baseada na teoria dos jogos e nos valores de Shapley. A teoria dos valores de Shapley distribui um "pagamento" a cada "jogador" de acordo com sua contribuição para o resultado do jogo. O SHAP trata cada característica como um jogador. A contribuição de cada característica é determinada calculando como sua presença em diferentes conjuntos de características afeta a saída do modelo.

Sendo x_i a i-ésima característica do modelo, o problema central é determinar o quanto x_i contribui para a predição $f(x)$. A contribuição marginal de x_i expressa o quanto ela adiciona ao resultado do jogo quando se junta a uma coalizão S, subconjunto do total de características F. A contribuição marginal de cada característica é dada por

$$\Delta_i(S) = f(S \cup \{x_i\}) - f(S). \tag{2.9}$$

O valor de Shapley para x_i, denotado por ψ_i, é a média de sua contribuição marginal sobre todas as permutações possíveis de características, conforme

$$\psi_i = \sum_{S \subseteq F \setminus \{x_i\}} \frac{|S|!(M - |S| - 1)!}{M!} \Delta_i(S). \tag{2.10}$$

Como a maioria dos modelos não pode ser executada com características ausentes, o termo $f(S)$ é aproximado usando o valor esperado condicional da predição, como em $f(S) \approx \mathbb{E}[f(x) \mid x_S]$. Isso leva à definição formal dos valores SHAP, expressa por

$$\psi_i(f, x) = \sum_{S \subseteq F \setminus \{x_i\}} \frac{|S|!(M - |S| - 1)!}{M!} \cdot \Big[\mathbb{E}[f(x) \mid x_{S \cup \{x_i\}}] - \mathbb{E}[f(x) \mid x_S]\Big]. \tag{2.11}$$

O cálculo exato dos valores SHAP é computacionalmente caro, com complexidade de $O(2^M)$. Para contornar isso, o KernelSHAP aproxima esses valores utilizando o *framework* de regressão linear do LIME, substituindo o *kernel* de ponderação heurístico do LIME por um *kernel* derivado da teoria dos valores de Shapley. O termo de regularização é, portanto, eliminado. O *kernel* de ponderação do KernelSHAP atribui pesos maiores a coalizões com quantidades de características próximas a 0 ou M, dado por

$$\pi_{\tilde{x}}(b) = \frac{M - 1}{\binom{M}{|b|} |b| \, (M - |b|)}. \tag{2.12}$$

O KernelSHAP calcula os valores SHAP (ψ_i) ao otimizar a função de perda

$$\mathcal{J}(f, g, \pi_{\tilde{x}}) = \sum_{b} \pi_{\tilde{x}}(b)[f(h_x(b)) - g(b)]^2, \tag{2.13}$$

que espelha a do LIME, mas com o *kernel* teoricamente fundamentado do SHAP.

Por construção, o KernelSHAP satisfaz propriedades desejáveis como acurácia local ($g(b) = f(x)$ quando $x = h_x(b)$), ausência ($b_i = 0 \implies \psi_i = 0$) e consistência (se a contribuição de uma característica aumenta, seu valor SHAP não pode diminuir).

2.5.3 Métricas de Avaliação

Em problemas de classificação, o desempenho de modelos é medido com base em acertos das decisões dentre um conjunto finito e discreto de classes dos dados. A ferramenta primária para extrair e visualizar essas informações é a matriz de confusão, uma estrutura tabular que cruza as predições realizadas pelo modelo com os rótulos reais do conjunto de dados. Em um cenário de classificação binária, essa matriz é composta por quatro elementos: os Verdadeiros Positivos (VP) e os Verdadeiros Negativos (VN), que representam os acertos do modelo nas respectivas classes, e os Falsos Positivos (FP) e Falsos Negativos (FN), que quantificam as predições incorretas [93].

A distinção entre falsos positivos e falsos negativos é de suma importância no contexto de detecção de anomalias de rede, já que revela se o sistema deixa de perceber tráfego espúrio ou pune injustamente usuários legítimos. O falso positivo, também conhecido como erro do Tipo I, ocorre quando o modelo sinaliza a presença de uma anomalia que não existe, funcionando como um alarme falso. Por outro lado, o falso negativo, erro do Tipo II, acontece quando o modelo classifica uma amostra positiva como negativa [94].

A partir dessas métricas primárias, calculam-se as taxas de erro específicas que ajudam a entender a proporção das falhas do modelo. A Taxa de Falso Positivo (TFP) mede a proporção de amostras que são reais negativas, mas que foram incorretamente rotuladas como positivas, sendo definida por

$$TFP = \frac{FP}{FP + VN}. \tag{2.14}$$

A Taxa de Falso Negativo (TFN) indica a proporção de observações positivas que o modelo não conseguiu identificar, calculada por

$$TFN = \frac{FN}{FN + VP}. \tag{2.15}$$

Para avaliar a confiabilidade do classificador em suas predições positivas, recorre-se comumente à Revocação e à Precisão. A Revocação, também chamada de Sensibilidade

ou Taxa de Verdadeiros Positivos, avalia a capacidade do modelo de encontrar todas as amostras que são de fato positivas, dada por

$$\text{Revocação} = \frac{VP}{VP + FN}. \tag{2.16}$$

A Precisão revela a proporção de acertos positivos dentre tudo aquilo que o modelo classificou como positivo, demonstrada em

$$\text{Precisão} = \frac{VP}{VP + FP}. \tag{2.17}$$

Com o intuito de resumir o *trade-off* entre precisão e revocação em um único número, utiliza-se frequentemente a métrica F1-Score, que consiste na média harmônica entre ambas [95]. Essa métrica é expressa em

$$\text{F1-Score} = 2 \times \frac{\text{Precisão} \times \text{Revocação}}{\text{Precisão} + \text{Revocação}} = \frac{2VP}{2VP + FP + FN}. \tag{2.18}$$

Contudo, o F1-Score desconsidera os Verdadeiros Negativos (VN), frequentemente presentes em dados desbalanceados como os de tráfego de redes de computadores. Nesses dados, o F1 pode apresentar uma avaliação enganosa, falhando em refletir a verdadeira competência do modelo ao ignorar os acertos da classe majoritária.

Para mitigar as limitações do F1, o Coeficiente de Correlação de Matthews (*Matthews Correlation Coefficient*, MCC) é adotado como uma alternativa mais confiável e robusta [96, 97]. O MCC leva em consideração todos os quadrantes da matriz de confusão e calcula um coeficiente de correlação entre os valores observados e preditos. O resultado contínuo varia de -1 (total discordância) a $+1$ (predição perfeita), passando por 0 (predição equivalente ao acaso). Essa métrica tende a ser mais confiável sob classes desbalanceadas e é definida por

$$\text{MCC} = \frac{(VP \times VN) - (FP \times FN)}{\sqrt{(VP + FP)(VP + FN)(VN + FP)(VN + FN)}}. \tag{2.19}$$

3 TRABALHOS RELACIONADOS

Sistemas de Detecção de Intrusão baseados em Aprendizado de Máquina e Aprendizado Profundo têm sido explicados por algoritmos de Inteligência Artificial Explicável buscando aumentar sua confiabilidade. Essas estratégias comumente atribuem importâncias às características dos dados de entrada, aumentando o entendimento sobre as inferências dos modelos e potencialmente possibilitando otimização transparente.

Al-Essa *et al.* [34] propuseram um método de detecção de ataques de rede nomeado PANACEA baseado em aprendizado por agrupamento e treino adversário. Os autores treinaram múltiplos modelos de rede neural profunda independentemente em quatro conjuntos de dados: NSL-KDD, UNSW-NB15, CIC-IDS2017 e CICMalDroid20. Os dados foram aumentados utilizando aprendizado adversário para gerar modelos mais robustos. DALEX, um *framework* de inteligência artificial explicável que combina explicadores baseados em LIME, SHAP, ALE e métodos de extração de conhecimento, foi empregado para determinar as importâncias das características para cada modelo. Essas importâncias foram agrupadas em *clusters* de acordo com sua similaridade em relação às características. Os autores selecionaram modelos com explicações distintas a fim de promover diversidade em aprender padrões de ataques variados. Ainda que, em média, o F1-Score ponderado tenha aumentado 5 pontos percentuais (p.p.) com o emprego do PANACEA, a ausência de transparência do modelo final foi apontada como uma limitação. Outra limitação é a falta de justificativa direta e explicável sobre a seleção de características.

Arreche *et al.* [35] desenvolveram um IDS com uma estrutura hierárquica de treino. Uma coleção de classificadores base é usada para melhorar a detecção multiclasse do sistema para ataques dos conjuntos de dados CIC-IDS2017, NSL-KDD e RoEduNet-SIMARGL2021. No primeiro nível da hierarquia, os autores empregam técnicas que vão de Árvores de Decisão até Redes Neurais Profundas. Esses modelos são alimentados a partir de duas estratégias de seleção de características: o conjunto completo e um subconjunto reduzido das oito mais impactantes identificadas pelo uso de explicações de SHAP em um estudo prévio. Esse estudo anterior baseou-se em métricas quantitativas, perturbações locais e os valores médios absolutos de SHAP, o que ignora a orientação das influências das variáveis. O segundo nível da hierarquia compreende múltiplos métodos de agrupamento treinados com as saídas dos modelos do primeiro nível, utilizando tanto as classes preditas quanto as distribuições de probabilidade. Esse estágio incorpora seleção de características baseada em Ganho de Informação para refinar os sinais preditivos mais significativos, aumentando o F1-Score do sistema em média em 2 p.p. Apesar de usar características selecionadas pelo SHAP como alternativa de treino dos modelos do primeiro nível, o melhor modelo agrupado final foi obtido considerando o conjunto completo de características.

Irénée *et al.* [31] propuseram um *framework* para melhorar o desempenho e a explicabilidade de um sistema classificador de tráfego HTTPS de duas camadas baseado em XGBoost. Os autores focam na detecção de ataques que tiram proveito das consultas DNS criptografadas por HTTPS, o *DNS over HTTPS* (DoH), usando o menor número possível de características do conjunto de dados CIRA-CIC-DoHBrw-2020. A primeira camada do sistema classifica o tráfego HTTPS, separando DoH das demais atividades. A segunda camada classifica o DoH como legítimo ou espúrio. As importâncias das características do modelo são analisadas usando TreeSHAP. O módulo de Avaliação Sequencial Progressiva iterativamente avalia todas as combinações de subconjuntos das dez características mais importantes em ambas as camadas, selecionando um conjunto ótimo de três e cinco características do conjunto original de 33, respectivamente. Apesar de os autores empregarem SHAP, a mistura e teste sistemático de múltiplas combinações de características prejudica a explicabilidade e torna o processo de seleção opaco. As explicações locais de SHAP analisadas pelos autores baseiam-se em quatro amostras aleatórias do conjunto de dados, cada uma representando um tipo de tráfego: DoH, não-DoH, DoH legítimo e DoH espúrio. O mesmo desempenho (*i.e.*, F1-Score de 100%) foi atingido com as 10 características mais importantes e a variante de menor dimensão; no entanto, a última apresentou uma diminuição de cerca de 70 milissegundos no tempo de inferência de cada camada.

Chinu e Bansal [36] implementaram um *framework* híbrido nomeado Fair-XIDS para melhorar o desempenho de detecção de classificadores de Floresta Aleatória, Árvore de Decisão, Máquina de Vetor de Suporte e Redes Neurais. O sistema foi avaliado em dois conjuntos de dados da plataforma Kaggle (*Malware Detection* e *Cyber Security Attacks*) e dois amplamente utilizados: CIC-IDS2017 e NSL-KDD. Eles empregaram o DTO-SMOTE para realizar sobreamostragem e tratar desbalanceamento de dados. Desenviesamento adversário foi usado durante o treino para neutralizar vieses algorítmicos. Finalmente, os autores aplicaram Classificação com Rejeição durante o pós-processamento para evitar que os modelos tomem decisões sob baixa confiança ou potencial viés. O desenviesamento adversário é calculado subtraindo o termo de regularização dos valores de SHAP, visando minimizar a perda preditiva e maximizar a perda adversária. O aspecto explicativo do XAI só é utilizado para explicar o modelo globalmente usando os valores médios absolutos de SHAP para cada característica, o que desconsidera impactos positivos e negativos necessários para avaliar vieses e características prejudiciais. Os autores destacam que o *framework* deles balanceia a compensação entre acurácia e equidade reduzindo a taxa de falsos positivos em média em 85%.

Shtayat *et al.* [37] propuseram um sistema de detecção de intrusão para o ambiente de Internet das Coisas Industrial que combina Redes Neurais Convolucionais, Votação Majoritária e XAI. Os autores implementaram três variações de CNNs com diferentes quantidades de camadas e nós. Esses modelos são combinados em um comitê, no qual as predições individuais passam por uma votação majoritária para definir a decisão fi-

nal. As três variações de CNN e o comitê são comparados com um modelo de Máquina de Aprendizagem Extrema e outros sistemas do estado da arte. Os classificadores são treinados para detecção de intrusão binária e multiclasse, tarefas nas quais o comitê atingiu métricas maiores para o conjunto de dados TON_IoT, com um F1-Score de 100% e 99,5%, respectivamente. As importâncias de características das categorizações binária e multiclasse foram explicadas pelos algoritmos de LIME e SHAP, exibindo relevâncias conflitantes. Apesar dessa inconsistência, inexplorada pelo trabalho, o carimbo temporal do pacote é demarcado como uma característica-chave da detecção. Esse comportamento, clarificado pelo método de XAI, pode indicar um viés de sobreajuste que os autores não detectaram ou mitigaram.

Mahmoud *et al.* [32] propuseram um sistema de detecção de intrusão explicável de dois estágios que combina classificação binária e multiclasse. O sistema inicialmente classifica tráfego como legítimo ou espúrio usando diversos modelos, indo de *Linear Support Vector Machine* (LinearSVM) até *Light Gradient Boosting Machine* (LightGBM). Posteriormente, modelos Bi-LSTM e CNN-Bi-LSTM são treinados somente com ataques dos conjuntos de dados CIC-IDS2017 e UNSW-NB15, originais e sobreamostrados, para classificá-los em categorias específicas. A técnica LightGBM foi selecionada por conta da sua baixa taxa de falsos negativos e alto F1-Score comparado com os demais modelos. Essa técnica foi, ainda, explicada globalmente por meio de SHAP. O modelo Bi-LSTM combinado com SMOTE foi designado como o classificador multiclasse final do *framework*. O sistema obteve 99,8% de F1-Score para o conjunto CIC-IDS2017 e 97,2% para o UNSW-NB15. Ainda que os autores comentem sobre a orientação das importâncias de SHAP com gráficos de enxame, potenciais vieses como a dominância da característica *Time-to-Live* não foram investigados.

Keshk *et al.* [33] implementaram um sistema de detecção de intrusão baseado em LSTM para classificação binária de tráfego tirando proveito de quatro técnicas de explicação: SHAP, *Permutation Feature Importance*, *Individual Conditional Expectations* e *Partial Dependence Plots*. A estrutura proposta no artigo explica o processo de decisão do sistema local e globalmente, analisando as características mais importantes a fim de aumentar sua confiabilidade. Os autores sugerem que métodos globais de XAI podem auxiliar no processo de seleção de características. No entanto, a metodologia proposta envolve a seleção das 20 características mais influentes de cada método, gerando subconjuntos de atributos oriundos de técnicas distintas que frequentemente divergem entre si. Em seguida, avalia-se a eficácia dessas seleções na detecção de ataques utilizando as bases de dados UNSW-NB15, TON_IoT e NSL-KDD. As características mais importantes apontadas pelo PFI e SHAP são combinadas, criando um subconjunto de dados que, ao treinar o sistema proposto, atinge métricas de detecção similares às dos modelos treinados com subconjuntos extraídos apenas baseando-se em um método de XAI. O F1-Score variou de 77% para o conjunto de dados NSL-KDD até 88% para o UNSW-NB15.

A maior parte dos trabalhos revisados que empregam XAI utiliza, especificamente, o SHAP para explicar a importância das características dos modelos. Frequentemente os métodos de explicação são subutilizados ao deixar de analisar as orientações das importâncias de características, comportamentos temporais e específicos de cada classe. Poucos artigos apresentam um uso alternativo de técnicas de explicabilidade para otimizar o desempenho dos modelos, como poda de comitês, cálculo de termos matemáticos e seleção de características. Essa escassez deve-se, em grande parte, à complexidade analítica de traduzir interpretações qualitativas em diretrizes sistemáticas de melhoria, somada ao alto custo computacional de processar explicações densas em larga escala. A Tabela 1 expõe a comparação dos trabalhos correlatos quanto ao emprego de estratégias de otimização baseadas em XAI e ao nível de transparência dessas abordagens, verificando se a otimização é completamente justificada pela análise das explicações do modelo.

Os estudos que exploram otimização baseada em Inteligência Artificial Explicável frequentemente manipulam as elucidações de forma a tornar o processo opaco, como ao combinar subconjuntos de características sem justificativa lógica. Essa prática enfraquece a premissa da explicabilidade, dificultando a detecção de vieses e a proposição de combinações de atributos aplicáveis de maneira global. Além disso, a dependência de subconjuntos de dados escolhidos aleatoriamente ou de forma empírica introduz riscos adicionais de viés pela ausência de uma estratégia robusta de seleção de amostras.

Para superar essas limitações, este trabalho propõe uma análise aprofundada de dois sistemas de detecção de intrusão, um baseado em Redes Adversárias Generativas e outro em Autocodificador Variacional β, interpretados sob a ótica do SHAP. Adota-se um método não supervisionado de subamostragem para preservar a representatividade dos dados e reduzir o custo computacional de XAI. As explicações isolam as classes de tráfego para que sejam avaliadas separadamente, permitindo identificar impactos específicos de cada categoria. O comportamento temporal do modelo é integrado à análise para guiar a seleção transparente de características sob a dinamicidade do ambiente de redes de computadores.

Tabela 1 – Comparação dos artigos do estado da arte que utilizam XAI como explicação e/ou otimização com este trabalho. **Fonte:** elaboração do próprio autor.

Trabalho	Métodos de XAI usados	Propõe otimização?	Otimização transparente?
Al-Essa *et al.* [34]	DALEX	Sim	Não
Arreche *et al.* [35]	SHAP	Sim	Não
Irénée *et al.* [31]	SHAP	Sim	Não
Chinu e Bansal [36]	SHAP	Sim	Não
Shtayat *et al.* [37]	SHAP e LIME	Não	-
Mahmoud *et al.* [32]	SHAP	Não	-
Keshk *et al.* [33]	SHAP, PFI, ICE e PDP	Sim	Não
Este trabalho	**SHAP**	**Sim**	**Sim**

4 OTIMIZAÇÃO TRANSPARENTE DE SISTEMAS DE DETECÇÃO DE ANOMALIAS

Neste capítulo, descreve-se a metodologia proposta para otimização transparente de sistemas de detecção de anomalias. Esta é dividida em cinco partes, como mostrado na Figura 5: Sistema de Detecção de Intrusão, Módulo de Explicação, Análise por Classes de Tráfego, Seleção de Atributos e Treinamento do Sistema. O sistema de detecção de intrusão é explicado sob a ótica do SHAP pelo Módulo de Explicação. As elucidações do algoritmo de XAI são analisadas separadamente por categoria de tráfego.

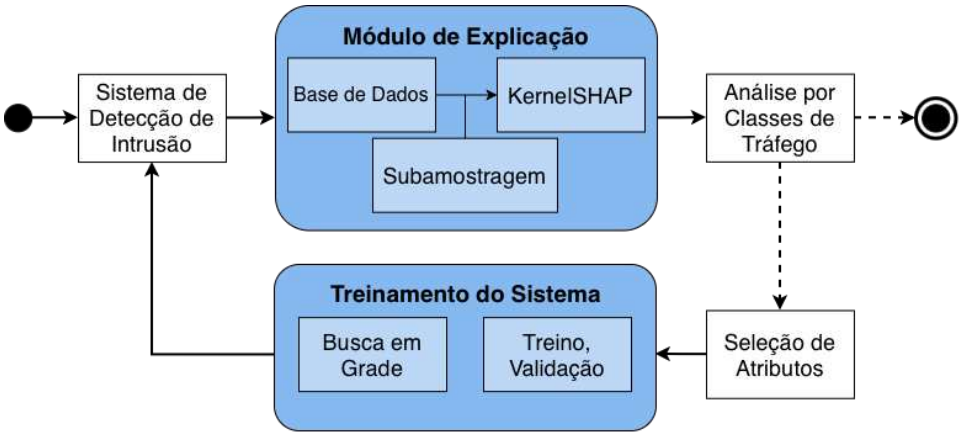

Figura 5 – Metodologia Proposta. **Fonte:** elaboração do próprio autor.

O fluxo metodológico segue um critério de refinamento baseado na elucidação das decisões do modelo. Caso não seja observado nenhum viés relacionado às características, o processo chega ao fim, produzindo um NIDS explicável. Caso contrário, os atributos que contribuem de forma indesejada na inferência do modelo são descartados e o sistema é retreinado utilizando o módulo de Treinamento do Sistema, o que gera uma nova instância de NIDS. O ciclo de explicação, seleção de atributos e treinamento do sistema é repetido até que não sejam detectadas características enganosas.

Os módulos de Explicação e de Treinamento do Sistema são subdivididos em etapas. O primeiro compreende a Base de Dados, Subamostragem e KernelSHAP. Neste módulo, o cenário de dados passa por um processo de subamostragem para reduzir a quantidade de instâncias que serão analisadas pelo algoritmo de KernelSHAP. Essa amostragem é necessária uma vez que a demanda de recursos computacionais do SHAP cresce linearmente com a quantidade de amostras consideradas na explicação. Esse subconjunto de dados deve ser representativo para que, de forma computacionalmente viável, as explicações sejam o mais fidedignas possível. Para isso, este trabalho propõe um algoritmo de subamostragem não supervisionado baseado em clusterização.

O Módulo de Treinamento do Sistema gera uma nova instância do NIDS utilizando os atributos selecionados a partir do conjunto de dados original. Esta etapa não se restringe ao simples ajuste do modelo a essas novas características, mas incorpora também uma busca em grade (*grid search*) para a otimização dos hiperparâmetros. Esse procedimento assegura a identificação da melhor configuração dentro do espaço de busca avaliado, atingindo um ótimo local. Contudo, a solução é considerada subótima globalmente, posto que é computacionalmente inviável testar todas as combinações paramétricas possíveis. Dessa forma, o NIDS é atualizado e calibrado para modelar com maior precisão o novo espaço de características.

4.1 Sistema de Detecção de Intrusão

O estudo de caso primário deste trabalho é o IDS proposto por Ruffo *et al.* [98]. A escolha de otimizar um sistema previamente validado e publicado em revista internacional visa extinguir o viés do autor. Ao separar a proposição da arquitetura de sua subsequente otimização, o trabalho oferece uma avaliação mais imparcial e rigorosa, garantindo que os resultados obtidos não sejam influenciados por decisões de projeto prévias. No trabalho original, os autores comparam três modelos de rede neural para construir o gerador e discriminador de uma Rede Adversária Generativa: Memória de Longo Curto Prazo (*Long Short-Term Memory*, LSTM), Redes Neurais Convolucionais Unidimensionais (*One-dimensional Convolutional Neural Network*, 1D-CNN) e Redes Convolucionais Temporais (*Temporal Convolutional Network*, TCN). Essas variações de GAN são comparadas utilizando métricas quantitativas de classificação, tempo para predizer todo o conjunto de teste e propensão a colapso de modo.

Os sistemas foram implementados com base nas entropias dos endereços IP de origem e destino, bem como das portas de origem e destino calculadas com a granularidade de um segundo de tráfego. Essas estatísticas, calculadas a partir do tráfego observado no instante t por meio da Entropia de Shannon [99], são armazenadas em uma janela deslizante com duração de τ segundos. A cada segundo, a janela é atualizada, descartando o valor mais antigo e incorporando o mais recente. O objetivo dos autores é identificar ataques volumétricos, como Negação de Serviço (*Denial of Service*, DoS) e Negação de Serviço Distribuída (*Distributed Denial of Service*, DDoS), a partir de um treino que compreende apenas tráfego de usuários legítimos. A validação e o teste dos modelos são feitos com dados contendo ataques, nos quais a rede discriminadora é responsável por distinguir tráfego legítimo e espúrio.

Os resultados experimentais da investigação revelam que, por mais que a LSTM-GAN apresente métricas de classificação aproximadamente 1 p.p. maiores, a 1D-CNN-GAN é cerca de 70% mais rápida em predizer o conjunto de teste e menos propensa ao colapso de modo. Por esse motivo, a GAN baseada em 1D-CNN foi apontada pelos autores

como o melhor modelo dentre os analisados na tarefa de detecção de intrusão. Apesar de sua capacidade de modelagem de dados, a 1D-CNN-GAN é uma caixa-preta, o que limita o entendimento dessa solução de segurança.

Esse NIDS é parte de um fluxo de trabalho operacional contínuo estabelecido no sistema original, como ilustrado na Figura 6. Conforme a arquitetura proposta, o processo se inicia com a etapa de Coleta de Tráfego, na qual os pacotes brutos que trafegam na rede são continuamente capturados e monitorados. Em seguida, os dados transitam para a fase de Pré-processamento, responsável por estruturar as informações e calcular as estatísticas de entropia nas janelas deslizantes de τ segundos. Uma vez formatadas, essas características alimentam o sistema de detecção de intrusão, no qual o modelo de aprendizado atua para distinguir padrões de uso legítimo de espúrio. Caso uma anomalia volumétrica seja detectada pelo modelo, a arquitetura avança para a fase de Mitigação, que orquestra as contramedidas de segurança e resulta no imediato bloqueio de tráfego espúrio, interrompendo a ação maliciosa e preservando a disponibilidade e qualidade de serviço da rede.

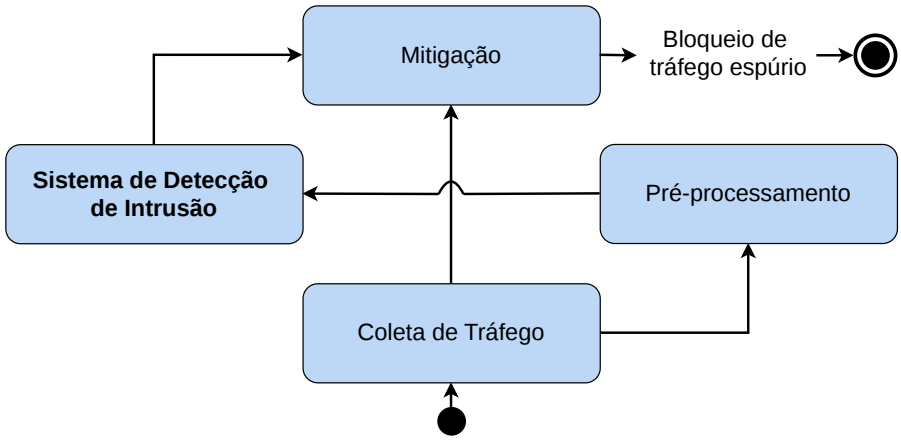

Figura 6 – Fluxo de trabalho. **Fonte:** adaptado de Ruffo *et al.* [98].

Adicionalmente, a fim de validar a metodologia de otimização proposta neste trabalho em outra arquitetura de rede neural, implementou-se um IDS baseado em Autocodificadores Variacionais β. Esse modelo segue um fluxo de trabalho análogo ao estabelecido por Ruffo *et al.* e também processa o tráfego com base nas entropias de Shannon com granularidade de um segundo; contudo, a análise é realizada de forma individual para cada segundo, dispensando o uso de uma janela deslizante para o armazenamento de registros históricos.

4.2 Módulo de Explicação

O Módulo de Explicação tem como objetivo esclarecer o funcionamento do sistema de detecção de intrusão e é composto pela base de dados, sua respectiva subamostragem e pela aplicação de inteligência artificial explicável.

4.2.1 Conjuntos de Dados

O conjunto de dados utilizado para o treinamento, validação e teste do estudo de caso principal é o CIC-DDoS2019, proposto pelo Instituto Canadense de Segurança Cibernética [38]. Adicionalmente, com o intuito de validar a metodologia de otimização do NIDS, utilizaram-se outros dois conjuntos de dados públicos: o CSE-CIC-IDS2018 [39] e o Orion2026 [40].

4.2.1.1 CIC-DDoS2019

A coleção de dados CIC-DDoS2019 [38] compreende dois dias de tráfego sintético organizado em Fluxos IP, nomeados dias de treino e teste. A coleta do dia de treino tem início às 10h30 e fim às 17h15, contendo 12 tipos distintos de ataques de negação de serviço distribuído, executados em intervalos separados. O dia de teste tem a coleta iniciada às 9h18 e finalizada às 17h36, contendo seis tipos de ataques de negação de serviço: NetBIOS, LDAP, MSSQL, UDP, UDP-Lag e SYN. Além desses ataques, o dia de teste inclui um ataque de mapeamento de portas, nomeado PortMap.

O trabalho de Ruffo *et al.* [98] pré-processa o dia de treino, removendo atividades maliciosas. Eles também excluem os ataques PortMap, UDP-Lag e SYN do dia de teste, dado que esses não expressam mudança volumétrica nas estatísticas de entropia.

4.2.1.2 CSE-CIC-IDS2018

O conjunto de dados CSE-CIC-IDS2018 [39], proposto pelo Instituto Canadense de Segurança Cibernética em parceria com a Amazon AWS, compreende 10 dias de tráfego de rede. Diferentemente do CIC-DDoS2019, este conjunto não segrega os dados previamente em partições de treinamento e teste, estruturando-os cronologicamente em dias da semana, todos contendo ataques injetados artificialmente. Durante o período analisado, apenas três variações de ataques de inundação volumétrica estão presentes: DoS Hulk, DDoS-LOIC-UDP e DDoS-HOIC.

O ataque DDoS-LOIC-UDP presente neste cenário não demonstrou comportamento volumétrico espúrio, mantendo as entropias próximas à linha de base das amostras benignas. Uma hipótese para esse comportamento reside no perfil do tráfego espúrio injetado, o qual mantém um padrão de envio de pacotes distribuído temporalmente. Portanto,

por não se destacar na codificação dos dados baseada em entropia, esse ataque foi retirado dos experimentos de teste.

Ademais, o sistema de detecção de intrusão foi treinado exclusivamente com as amostras legítimas extraídas da quinta-feira, 15 de fevereiro, originalmente populada por ataques de negação de serviço lentos. A validação do modelo ocorreu com o tráfego da sexta-feira, 16 de fevereiro, contendo o ataque DoS Hulk, enquanto a fase de teste foi realizada com os dados da quarta-feira, 21 de fevereiro, referentes ao ataque DDoS-HOIC.

4.2.1.3 Orion2026

O *dataset* Orion2026 [40], proposto pelo grupo de pesquisa Orion da Universidade Estadual de Londrina, é composto pelo tráfego emulado de cinco dias de coleta. Os dois primeiros dias contêm exclusivamente dados legítimos, direcionados ao treinamento e à validação do sistema. Os três dias subsequentes introduzem oito categorias de ataques de negação de serviço em ordens distintas, abrangendo diferentes protocolos de rede, como UDP, ICMP e variações de sinalizadores do protocolo TCP (SYN, ACK, PSH, URG, FIN e RST). Cada um desses dias remanescentes possui uma variação na cardinalidade entre atacantes e vítimas: o terceiro dia adota o cenário de um atacante para uma vítima (1-1); o quarto dia expande para N atacantes visando uma única vítima (N-1); e o quinto dia apresenta múltiplos atacantes contra múltiplas vítimas (N-N).

O sistema treinado no cenário Orion2026, utilizando o conjunto original de quatro características de entropia, foi avaliado nos dias que contêm tráfego espúrio. O ataque baseado em ICMP, conhecido como inundação de *ping*, foi desconsiderado nesta análise por gerar poucos fluxos IP e causar baixo impacto global nos atributos de entropia. Por não alterar o perfil volumétrico da rede, esse tipo de negação de serviço torna-se indetectável pela modelagem volumétrica abordada.

4.2.2 Amostragem Baseada em Clusterização por Densidade

O algoritmo SHAP apresenta um custo computacional que cresce de forma exponencial em relação à dimensionalidade dos dados e linearmente em relação ao tamanho do conjunto de dados submetido à explicação. Embora abordagens como o KernelSHAP busquem mitigar essa demanda computacional integrando conceitos do algoritmo LIME, o gerenciamento exaustivo de recursos de memória e processamento continua sendo um desafio crítico. Diante desse cenário, este trabalho propõe um método de subamostragem projetado para diminuir o volume de dados explicados, enquanto preserva sua representatividade estatística e explicabilidade. Essa redução é alcançada por meio de uma abordagem conjunta que emprega o HDBSCAN (*Hierarchical Density-Based Spatial Clustering of Applications with Noise*), o KDE (*Kernel Density Estimation*), a estimativa de Scott (*Scott's estimate*) [100] e a amostragem aleatória ponderada.

Essa estratégia busca superar limitações comuns em métodos tradicionais de subamostragem, como a seleção aleatória estratificada. Seleções aleatórias tendem a priorizar os dados estatisticamente mais frequentes, o que pode comprometer a diversidade dentro de uma mesma classe e levar à perda de variações dos dados, incluindo grupos inteiros de anomalias. O algoritmo proposto procura preservar a variedade intraclasse e reduzir a influência da aleatoriedade no processo de seleção, garantindo maior representatividade dos dados.

Variações de comportamento no tráfego podem segmentar os dados de uma mesma classe em agrupamentos distintos, possibilitando sua representação em *clusters*. Para o algoritmo proposto, a distribuição dos pontos de dados foi modelada com base em clusterização não supervisionada por densidade utilizando o algoritmo HDBSCAN separadamente para cada classe de dados. Essa abordagem permite a identificação da variação dos dados mesmo em regiões de menor densidade. O HDBSCAN calcula a distância de alcance mútuo entre pontos de dados para garantir que os *clusters* sejam definidos com base na densidade local, e não em limiares globais. A partir dessa distância, constrói-se hierarquicamente uma Árvore Geradora Mínima, unindo grupos conforme sua conectividade por densidade. Os agrupamentos são selecionados por meio de uma avaliação de estabilidade relativa, considerando aqueles que persistem em uma faixa mais ampla de densidades.

Contudo, o HDBSCAN depende de um hiperparâmetro principal, que codifica o número mínimo de pontos para formar um grupo. Um valor muito baixo dessa variável pode fragmentar os dados em excesso, considerando variações menores como ruído; um valor muito alto pode falhar em detectar subgrupos menores, tratando-os como parte de um mesmo *cluster*. Encontrar o equilíbrio é essencial para que os agrupamentos representem da melhor forma possível a estrutura dos dados.

Para determinar esse hiperparâmetro de forma automática, o algoritmo proposto emprega Otimização Bayesiana. Esse método é utilizado para encontrar o valor que maximiza a métrica não supervisionada *silhouette score*. Essa é uma medida que quantifica a qualidade dos *clusters*, avaliando o quão bem cada ponto de dado se encaixa em seu próprio *cluster* (coesão) em comparação com os *clusters* vizinhos (separação). A Otimização Bayesiana, por sua vez, testa iterativamente diferentes valores para o hiperparâmetro, aprendendo com os resultados anteriores para focar a busca nas configurações mais promissoras, evitando o custo computacional de uma busca exaustiva.

O KDE é executado para estimar a função de densidade probabilística de cada *cluster* de dados. O *kernel* utilizado foi o Gaussiano, cujo parâmetro de suavização λ é definido com base na estimativa de Scott para dados multivariados, expressa por $\lambda = n^{-1/(M+4)}$. Nessa equação, n representa o número de instâncias de dados contidas no *cluster* e M a dimensionalidade dos dados, *i.e.*, número de *features*. Esse passo agrega ao algoritmo informação da disposição dos pontos de dados dentro de um *cluster*. Finalmente, os pon-

tos de cada grupo recebem pesos de acordo com suas densidades probabilísticas e são selecionados com amostragem aleatória ponderada.

A estratégia proposta realiza a subamostragem com base na porcentagem de dados que se deseja manter. Neste trabalho, o tamanho da amostra foi definido pelo limite da capacidade operacional do ambiente de execução a fim de evitar esgotamento de memória (*Out of Memory*, OOM), restringindo o subconjunto a 600 instâncias do conjunto original. O conjunto subamostrado preserva tanto as nuances locais quanto globais dos dados, permitindo que o KernelSHAP produza explicações consistentes com aquelas obtidas a partir do *dataset* completo. As diferenças observadas entre diferentes tamanhos de subamostragem mostraram-se negligenciáveis, assegurando a viabilidade computacional do método sem comprometer a qualidade das explicações.

O Algoritmo 1 descreve o fluxo proposto, evidenciando as etapas desde a seleção do tamanho subótimo de *cluster* até a execução da amostragem ponderada por densidade.

Algoritmo 1 Subamostragem Proposta. **Fonte:** elaboração do próprio autor.

Entrada: Conjunto de dados tabular D contendo as *features* F e os rótulos das instâncias
Entrada: Taxa de amostragem α, limites de tamanho de *cluster* $[s_{min}, s_{max}]$
Saída: Conjunto subamostrado D'

Procedimento SUBAMOSTRAR(D, α, s_{min}, s_{max})
    $F \leftarrow$ *features* de D sem os rótulos
    // Etapa 1: Encontrar tamanho de cluster subótimo
    $s^* \leftarrow$ OtimizaçãoBayesiana$(D, F, [s_{min}, s_{max}])$
    // Etapa 2: Agrupar cada classe de ataque
    para cada classe Γ **em** D **faça**
        $D_\Gamma \leftarrow$ subconjunto padronizado de D para a classe Γ
        rótulos_clusters$_\Gamma \leftarrow$ HDBSCAN(D_Γ, s^*)
    fim para
    para cada *cluster* γ **em** rótulos_clusters$_\Gamma$ **faça**
        // Etapa 3: Calcular largura de banda ótima usando a fórmula de Scott
        $\lambda \leftarrow n^{-1/(M+4)}$
        // Etapa 4: Estimar densidade por cluster
        densidade$_\gamma \leftarrow$ KDE(γ, λ)
    fim para
    // Etapa 5: Realizar amostragem ponderada pela densidade
    para cada classe Γ **e** *cluster* γ **faça**
        $D'_{\Gamma,\gamma} \leftarrow$ amostrar α dos pontos ponderados por densidade$_\gamma$
        $D' \leftarrow D' \cup D'_{\Gamma,\gamma}$
    fim para
    retorne D'
fim Procedimento

4.3 Análise por Classes de Tráfego

As elucidações de SHAP revelam globalmente as importâncias das características ao unir explicações de múltiplas instâncias individuais. Contudo, a dependência de certas métricas quantitativas e agregadas para orientar a otimização do modelo apresenta limitações operacionais. Métricas de importância global, como a média dos valores absolutos de SHAP, tendem a achatar a informação, sumarizando apenas a magnitude do impacto e ocultando a direcionalidade e a distribuição dos dados ao longo das instâncias. O uso de métricas quantitativas de equidade (*fairness*) atua predominantemente como ferramenta para acusar o desequilíbrio de modelagem em que uma classe é penalizada. Dessa maneira, *fairness* não oferece a granularidade necessária para a intervenção no espaço de características.

Para contornar essas limitações e transpor a barreira do diagnóstico para uma engenharia de características, este trabalho adota a análise visual das distribuições de SHAP por meio de gráficos de enxame separados por classe de tráfego. Diferentemente de uma métrica quantitativa, essa abordagem gráfica preserva simultaneamente a densidade, a direcionalidade do impacto na predição e o valor real da característica. A extração dessas informações permite mapear a origem das falhas do modelo ao identificar visualmente características que induzem a classificações errôneas. Cenários como atributos que indevidamente elevam a pontuação de anomalia de registros legítimos (gerando falsos positivos) ou que atenuam essa mesma pontuação em instâncias de ataque (gerando falsos negativos) são melhor observados visual e qualitativamente.

4.4 Seleção de Atributos

O sistema de detecção de anomalias, treinado apenas com dados de tráfego de usuários legítimos, aprende a modelar o comportamento benigno da rede, detectando ocorrências que desviam do padrão estabelecido. Por esse motivo, deseja-se que os impactos das características do tráfego legítimo se aproximem do valor esperado de SHAP, enquanto os ataques devem apresentar influências divergentes do esperado. As características que não se alinham com o comportamento desejado são descartadas, sendo consideradas enviesadas e responsáveis por falsas classificações.

4.5 Treinamento do Sistema

O módulo de Treinamento do Sistema gera uma instância de IDS a partir dos atributos selecionados. Inicialmente, os pesos do modelo são configurados com valores aleatórios. O treino é então conduzido exclusivamente com dados de tráfego legítimo, em um processo iterativo baseado em propagação direta e retropropagação.

A propagação direta consiste em alimentar o modelo com uma instância de dados, que passa pelas camadas da rede até gerar uma inferência na saída. Em seguida, a retropropagação calcula o erro dessa inferência em relação ao valor correto (o rótulo legítimo) e propaga esse erro retroativamente pelas camadas da rede neural. Os pesos do modelo são atualizados para minimizar a divergência entre a saída e o rótulo verdadeiro.

Para diminuir o custo computacional e tempo relacionados às propagações de amostras individuais, uma abordagem de treino em lotes é adotada. Em vez de propagar e retropropagar amostras uma a uma, um lote de dados é processado na propagação direta. Somente após o lote completo, o erro médio é calculado e uma única atualização de pesos ocorre via retropropagação.

A propagação e a retropropagação completas do conjunto de dados de treino pela rede concluem uma época de treinamento. Para o modelo 1D-CNN-GAN, Ruffo *et al.* [98] propõem uma etapa de validação ao final de cada época, utilizando amostras que contêm ataques. Com o intuito de evitar o vazamento de dados (*data leakage*), esse subconjunto de validação é estritamente disjunto daquele usado na fase de teste. Durante cada validação, a arquitetura baseada em redes adversárias avalia o desempenho do modelo por meio da métrica MCC. Para isso, o limiar de decisão da saída do discriminador é variado no intervalo $[0{,}0; 0{,}3]$ por meio de 150 pontos de teste, o que corresponde a um incremento de aproximadamente 0,002. Seleciona-se o limiar que promove a melhor separação das classes, caracterizando o método como semissupervisionado devido ao uso de rótulos nessa fase.

Em contrapartida, a abordagem baseada em β-VAE detecta amostras espúrias valendo-se do erro de reconstrução dos segundos de tráfego codificados em entropias de Shannon. Esse modelo é treinado exclusivamente com dados legítimos e validado em um conjunto estritamente benigno. Para a definição do limiar de anomalia, adota-se a Teoria dos Valores Extremos (*Extreme Value Theory*, EVT), proposta por Fisher e Tippett em 1928 [101], na qual a parcela de 5% do tráfego que apresenta o maior erro de reconstrução é classificada como espúria. Como o método dispensa o uso de rótulos tanto no treinamento quanto na validação, ele é considerado não supervisionado.

Todo esse ciclo de treinamento e validação é encapsulado por uma busca em grade, encarregada de testar sistematicamente as combinações de hiperparâmetros em um espaço discreto e predefinido. Ao contrário dos pesos da rede, esses hiperparâmetros (*e.g.*, tamanho do lote (*batch size*) e o número de épocas) não são otimizados autonomamente durante o treinamento. Ao final da busca, a configuração semissupervisionada define seus parâmetros ideais com base no maior valor de MCC obtido na validação. Por outro lado, a arquitetura não supervisionada elege os parâmetros que resultam no menor erro de reconstrução perante o conjunto de validação benigno.

5 EXPERIMENTOS E RESULTADOS

Este capítulo exibe os resultados experimentais advindos da aplicação da metodologia proposta para explicar e otimizar IDS baseados em modelos caixa-preta. A avaliação é ancorada em um estudo de caso principal, empregando a arquitetura 1D-CNN-GAN no cenário de dados CIC-DDoS2019, e complementada por cenários alternativos para explorar a generalização do método em diferentes contextos de dados e algoritmos. Inicialmente, uma Análise Exploratória de Dados é conduzida no cenário principal, visando revelar as relações estatísticas das características e fundamentar a interpretação das decisões do modelo. Em seguida, aplica-se a estratégia de subamostragem não supervisionada, visando assegurar que a extração das elucidações do SHAP permaneça computacionalmente viável e representativa do tráfego original.

A etapa de explicabilidade avalia as elucidações do XAI de forma segmentada por classe de tráfego. Essa separação evita o mascaramento de dinâmicas globais que, se agregadas, potencialmente impossibilitariam a identificação da origem de comportamentos enviesados. Com base nessas distribuições visuais do SHAP, conduz-se um processo de engenharia de características transparente, removendo os atributos responsáveis por introduzir falsos positivos e falsos negativos nas predições do modelo. Esse ciclo de poda é iterado até a consolidação de um sistema estável, com lógicas de decisão contraditórias reduzidas. A versão otimizada dos sistemas é comparada com suas respectivas versões originais, balizando-se por métricas de classificação e desempenho computacional. Por fim, o modelo resultante é submetido a uma última auditoria de explicabilidade para reforçar sua confiabilidade na tarefa crítica de segurança.

Para avaliar a viabilidade operacional dos modelos otimizados, optou-se pela medição do tempo de inferência em detrimento de métricas teóricas, como a complexidade algorítmica (*Big O*) ou o custo computacional em Operações de Ponto Flutuante (*Floating-Point Operations*, FLOPs). Embora as métricas teóricas forneçam limites superiores assintóticos, elas falham em capturar os impactos da aceleração de *hardware* moderno, otimizações de *frameworks* e o paralelismo inerente a arquiteturas de rede neural. Além disso, no contexto de IDS, a viabilidade de uma solução é ditada pela sua capacidade de inspecionar o tráfego de rede em tempo real, tornando a medição do tempo de inferência uma métrica viável para atestar sua aplicabilidade em ambientes de produção.

5.1 Ambiente de Execução

Para promover a reprodutibilidade e a consistência na avaliação dos modelos propostos, todos os experimentos detalhados neste capítulo foram executados em um ambiente computacional local. O desenvolvimento e a avaliação do sistema de detecção de intrusão foram conduzidos na linguagem Python. Para a construção das arquiteturas de aprendizado de máquina, manipulação eficiente dos dados de tráfego de rede e explicação dos modelos caixas-pretas, foram utilizadas as bibliotecas *TensorFlow*, *NumPy*, *Pandas* e *shap*. As especificações detalhadas de *hardware* e as versões específicas dos *softwares* empregados estão sumarizadas na Tabela 2.

Tabela 2 – Especificações de *hardware* e *software* do Ambiente de Execução dos experimentos. **Fonte:** elaboração do próprio autor.

Recurso	**Especificação**
Processador	Intel Core i7-9750H 2,6 GHz
Memória RAM	16 GB DDR4 2666 MHz
Sistema Operacional	Ubuntu 22.04 LTS
Python	3.10.0
TensorFlow	2.15.0
NumPy	1.26.4
Pandas	2.2.0
shap	0.46.0

5.2 Análise Exploratória de Dados

Os dias de treino e teste pré-processados em entropias que são utilizados pelo sistema original têm suas distribuições exibidas nas Figuras 7 e 8, respectivamente. As variáveis de entropia são representadas por H, sendo H(Porta orig) referente à entropia de porta de origem, H(Porta dest) à entropia de porta de destino, H(IP orig) à entropia de endereço IP de origem e H(IP dest) à entropia de endereço IP de destino.

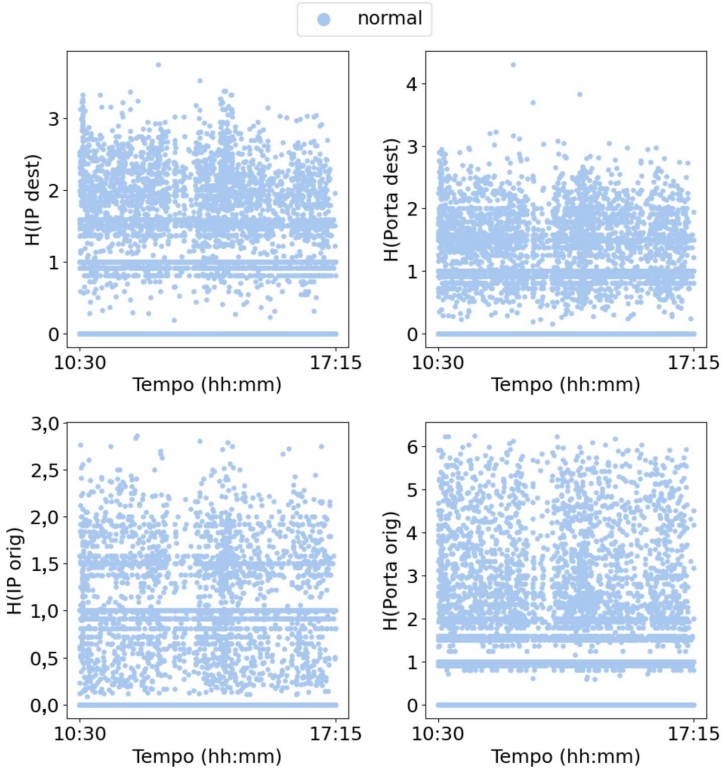

Figura 7 – Distribuição das amostras do dia de treino do conjunto de dados CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

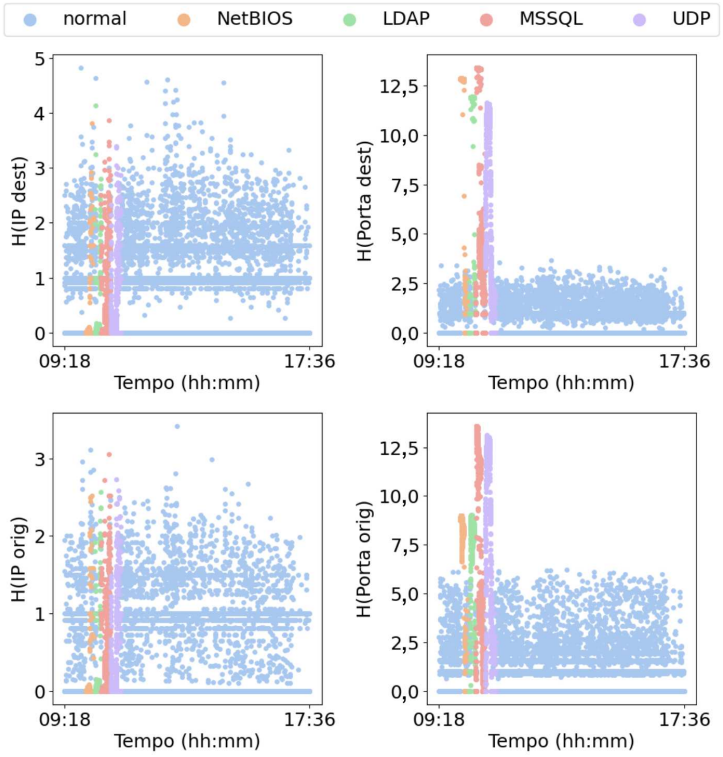

Figura 8 – Distribuição das amostras do dia de teste do conjunto de dados CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

A Figura 9 ilustra as relações par a par das variáveis de entropia e suas respectivas funções de densidade de probabilidade para diferentes categorias de tráfego. As relações das variáveis mostram interações significativas, apesar de geralmente se alinharem com os padrões esperados. Uma tendência diagonal positiva emerge entre as entropias de origem e destino correspondentes (*e.g.*, IP de origem e destino e porta de origem e destino). Esse comportamento é intuitivo, uma vez que uma variável de origem naturalmente influencia sua variável correspondente de destino quando um pacote flui do *host* de origem para o de destino.

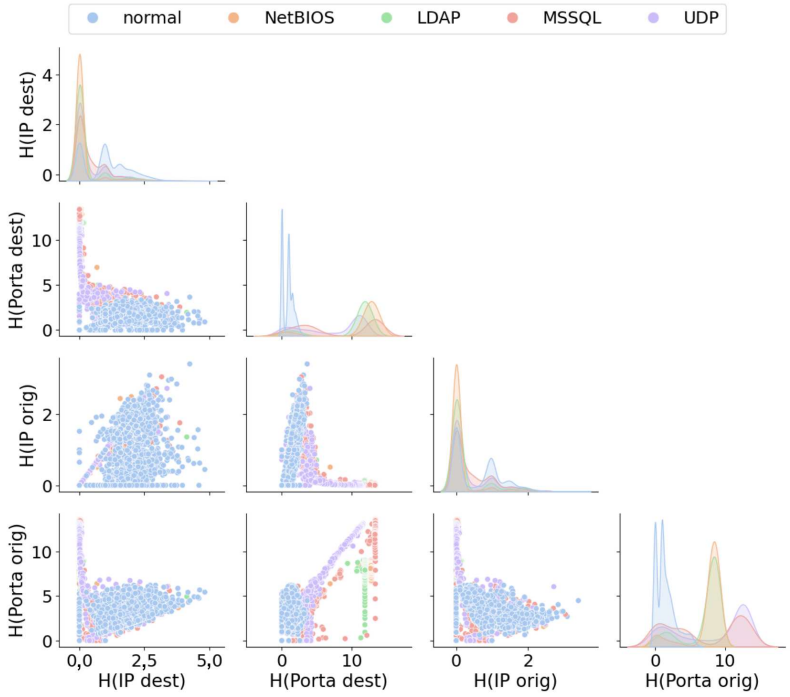

Figura 9 – Relações par a par das características do dia de teste do conjunto de dados CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

Ao comparar os endereços IP com as portas, observa-se um padrão de separação de dados em formato de "L". Tráfego legítimo se concentra em valores baixos de entropia de porta, variando ao longo de diversos intervalos da entropia de endereços IP. Diferentes ataques exibem uma distribuição mais espalhada em ambos os eixos. Essa dispersão indica que o tráfego malicioso apresenta menor regularidade e maior variabilidade em suas características, quando comparado ao tráfego legítimo. Tal fato pode ser atribuído à natureza diversa das táticas de ataque, as quais exploram protocolos e vulnerabilidades para comprometer um espectro de serviços em redes de computadores.

A densidade de probabilidade presente na diagonal principal da Figura 9 indica que as entropias de endereços IP do tráfego legítimo sobrepõem-se às do tráfego espúrio, concentrando-se em valores menores do espectro. O tráfego legítimo tende a agrupar-se em valores menores de entropias de porta, enquanto ataques de negação de serviço distribuído prevalecem em valores mais altos dessas entropias.

Cada tipo de ataque explora vulnerabilidades distintas; consequentemente, o tráfego gerado por cada um deles revela diferentes distribuições de entropia. A Figura 10 exibe as categorias de tráfego para cada variável de entropia por meio de um gráfico de violino. Esse recurso visual é uma variação do diagrama de caixa que incorpora a estimativa de densidade probabilística dos dados. Seções mais largas indicam uma maior probabilidade de encontrar observações naqueles intervalos, enquanto seções mais estreitas apontam para uma menor ocorrência. A análise da distribuição para os endereços IP de origem e de destino revela que a maioria das instâncias de ataque se concentra em valores mais baixos de entropia. No entanto, as medianas, denotadas por linhas horizontais com pontos pretos, e as regiões de pico probabilístico dos ataques sobrepõem-se a áreas de alta densidade do tráfego legítimo, um fator que dificulta a distinção entre eventos normais e espúrios.

Por outro lado, as entropias de portas exibem uma separação mais clara dessas categorias de tráfego. Para essas características, NetBIOS, LDAP e UDP apresentam maiores concentrações e medianas de entropia acima do limite superior do tráfego legítimo. A mediana do ataque MSSQL encontra-se próxima ao limite superior dos dados legítimos, e o ataque dispersa grandes concentrações de instâncias em valores tanto altos quanto baixos das características de entropia.

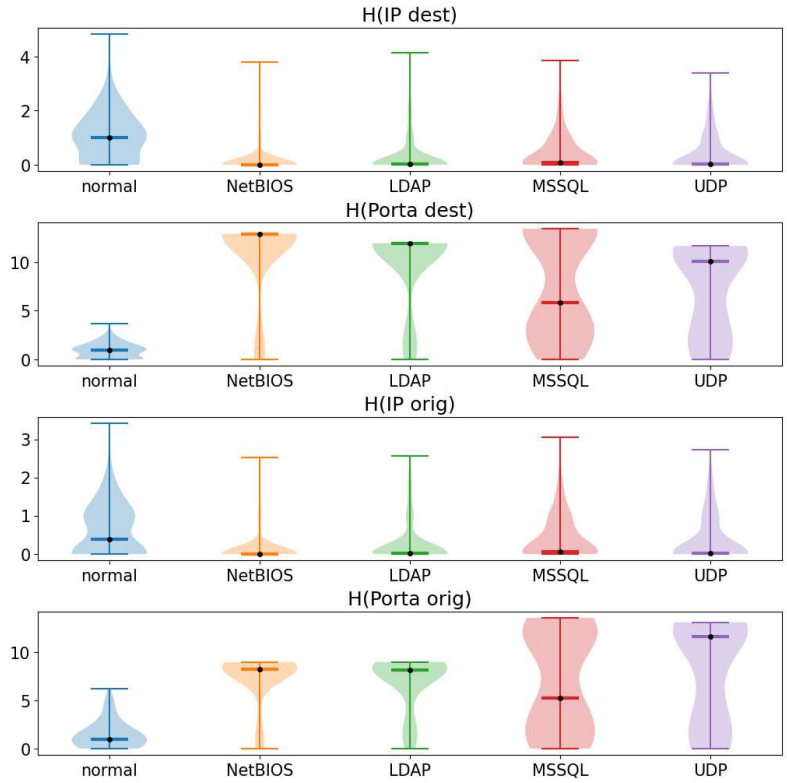

Figura 10 – Gráfico de violino das características do dia de teste do conjunto de dados CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

5.3 Subamostragem

Aplicando a estratégia de subamostragem proposta na Seção 4.2.2, uma versão menor do dia de teste do conjunto de dados CIC-DDoS2019 é obtida, retratada na Figura 11. Essa figura ilustra a distribuição probabilística dos dados do dia original e do dia subamostrado, utilizando cores mais claras e escuras, respectivamente. As distribuições para os dois dias apresentam formatos similares, preservando regiões de populações maiores e menores. Essa similaridade demonstra a capacidade do método proposto em manter a representatividade dos dados.

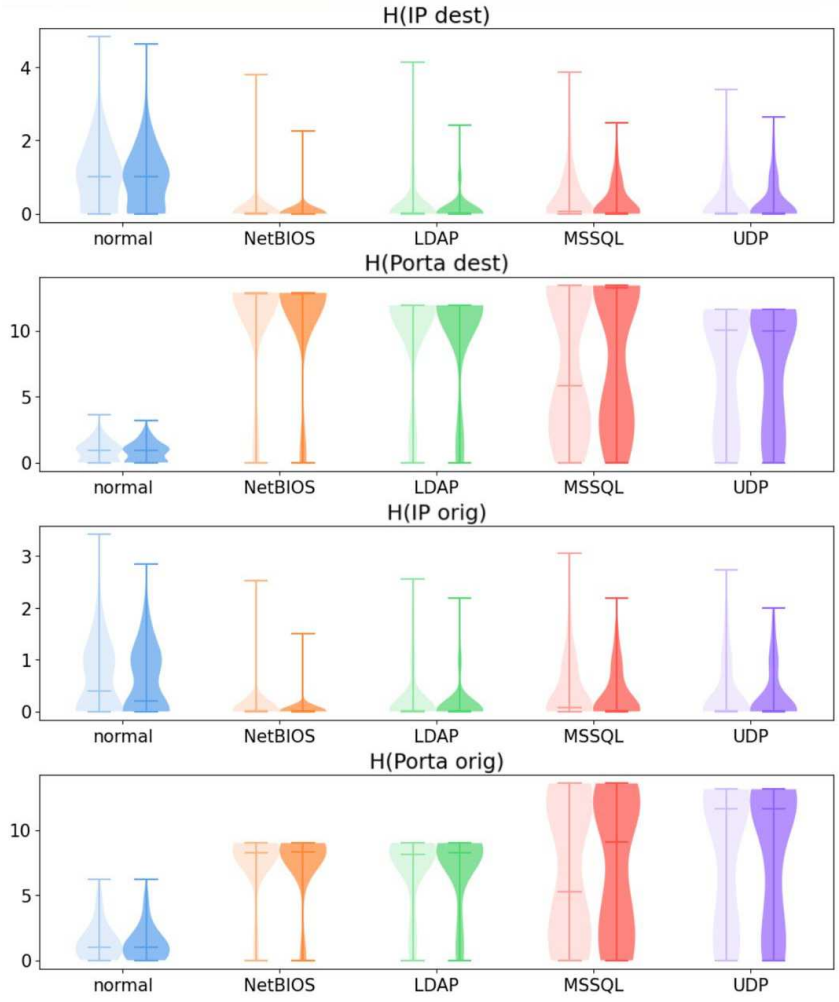

Figura 11 – Comparação entre as distribuições das características dos dias de teste original (cor clara) e subamostrado (cor escura) do conjunto de dados CIC-DDoS2019 em diferentes categorias de tráfego. **Fonte:** elaboração do próprio autor.

A Tabela 3 exibe a média e o desvio padrão dos dias de teste originais e subamostrados do *dataset* CIC-DDoS2019, separados por característica. É possível observar que a média e o desvio padrão diferem por erros médios absolutos de 0,18 e 0,14, respectivamente. Essas divergências indicam que as propriedades estatísticas dos dados foram

essencialmente preservadas, até mesmo na presença de categorias de tráfego distintas. Em casos como o do ataque MSSQL, a mediana pode variar, visto que essa classe é mais esparsa, como mostrado pela análise exploratória. Classes esparsas comumente possuem *clusters* em intervalos de dados distintos, afetando a mediana. Naturalmente, valores atípicos que não contribuem com a formação de agrupamentos tendem a ser deixados de fora das distribuições subamostradas. Esses acontecimentos explicam a variação na média e desvio padrão e a diferença nos limites superiores e inferiores, respectivamente.

Tabela 3 – Média e desvio padrão das características de entropia de tráfego nos conjuntos de teste original e subamostrado do CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

Característica	Conjunto Original (Média ± Desvio Padrão)	Conjunto Subamostrado (Média ± Desvio Padrão)
H(IP dest)	0,7708 ± 0,8479	0,6846 ± 0,7764
H(Porta dest)	3,6386 ± 4,7915	3,9619 ± 5,0165
H(IP orig)	0,4868 ± 0,6234	0,4155 ± 0,5695
H(Porta orig)	3,5692 ± 4,0759	3,8171 ± 4,2884

Uma comparação entre um método de subamostragem aleatório estratificado e a estratégia proposta neste trabalho é conduzida a fim de demonstrar a eficácia do método proposto. Ambas as técnicas foram executadas dez vezes para avaliar seus comportamentos não determinísticos. A distância média de Wasserstein entre os grupos de dados extraídos por ambos os métodos e os grupos de dados originais foi calculada.

A distância de Wasserstein quantifica o mínimo esforço necessário para transformar uma distribuição probabilística em outra. Sendo assim, distâncias menores indicam distribuições mais similares [102]. Essa métrica pode obscurecer a perda de diversidade ao favorecer regiões de dados mais densas. Para abordar essa limitação, o número de *clusters* mantidos para cada estratégia também é levado em conta nessa comparação, proporcionando maior clareza sobre suas estabilidades e capacidades de preservar variabilidade intraclasse.

A Tabela 4 compara as variações entre execuções da amostragem aleatória estratificada e da abordagem proposta neste trabalho. A distância média de Wasserstein entre agrupamentos de dados extraídos aleatoriamente e a distribuição original é de 0,0191 com um desvio padrão de 0,0018. Esse método manteve em média 50,2 *clusters*, com um desvio padrão de 3,29. Em contraste, a amostragem proposta apresentou uma distância média de Wasserstein de 0,0205 em relação aos dados originais, com uma variação entre execuções menor que a do modelo aleatório (desvio padrão de 0,0004). Essa abordagem preserva aproximadamente 51,3 agrupamentos em média, com um desvio padrão de 0,4714. Enquanto ambos os procedimentos tiveram similaridade distribucional comparável, o proposto neste trabalho consistentemente preservou mais *clusters* com menos variações entre execuções.

Tabela 4 – Comparação entre a distância de Wasserstein entre *clusters* e o número de *clusters* da amostragem aleatória estratificada e da subamostragem proposta. **Fonte:** elaboração do próprio autor.

Métrica	Amostragem Aleatória	Abordagem Proposta
Distância Média de Wasserstein	0,0191	0,0205
Desvio Padrão (Wasserstein)	0,0018	0,0004
Número Médio de *Clusters*	50,2	51,3
Desvio Padrão (*Clusters*)	3,2900	0,4714

Esses resultados foram obtidos no conjunto de dados CIC-DDoS2019; para outros cenários de dados com maior número de anomalias de tráfego ou maior sensibilidade a inicializações aleatórias, métodos baseados em amostragem estocástica são mais propensos a sofrer maiores inconsistências e perda de densidade estrutural. A técnica proposta neste trabalho é inerentemente mais estável e resiliente a essas variações, tornando-a uma solução mais confiável para preservar a heterogeneidade dos dados sob restrições de subamostragem.

5.4 Explicação Inicial

Os experimentos conduzidos para a explicação do sistema de detecção de anomalias baseiam-se na implementação oficial do KernelSHAP para a linguagem de programação Python. Esse método inicia um modelo de substituição baseado em regressão linear ponderada, ajustando-o aos dados de treino. O modelo de substituição é usado para computar os valores de SHAP para os dados de teste.

Uma vez que cada tipo de ataque explora diferentes vulnerabilidades, padrões distintos de tráfego emergem. Usar uma explicação única para todo o *dataset* gera constatações enganosas, já que mescla comportamentos potencialmente divergentes do modelo. Por esse motivo, a análise deste trabalho foi conduzida em gráficos separados por categorias de tráfego. Separando as classes de dados, é possível avaliar quais características são essenciais para que o modelo aprenda o comportamento legítimo da rede de computadores, e quais dificultam a detecção de ataques.

A Figura 12 mostra as explicações do modelo separadas por categoria de tráfego. O eixo vertical que intercepta o eixo x do gráfico de enxame no valor zero demarca a predição média do modelo para os dados de treino, chamada de valor esperado. Cada característica de entropia é representada no gráfico por um nome composto por duas partes principais: a informação da conexão de rede e o carimbo temporal. O carimbo temporal localiza a instância dentro da janela deslizante fornecida ao modelo 1D-CNN-GAN. Por exemplo, H(Porta dest)_t-7 denota a entropia de porta de destino sete segundos antes do tempo corrente. H(IP orig)_t-0 indica a entropia de endereço IP de origem no instante atual.

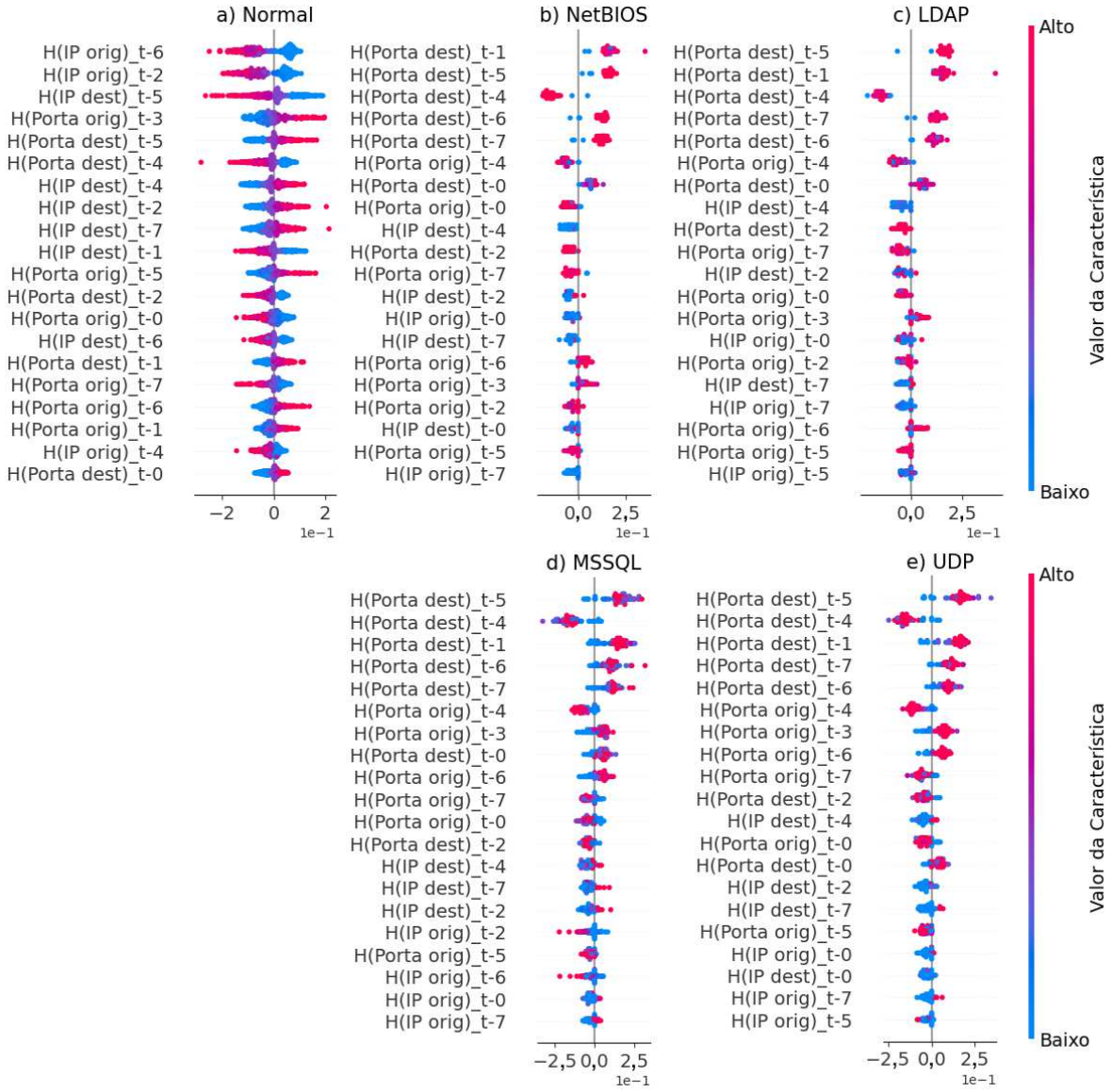

Figura 12 – Gráfico de enxame do dia de teste do conjunto de dados CIC-DDoS2019 separado por categoria antes da otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

No gráfico, cada ponto alinhado verticalmente a uma característica representa uma instância de dados. A cor do ponto ilustra o valor da característica, variando de azul a vermelho para valores baixos e altos. A posição do ponto ao longo do eixo x codifica a magnitude de variação que essa instância particular causou na saída do modelo em relação ao valor esperado. Pontos em valores negativos do eixo x indicam que a saída do modelo foi influenciada negativamente, evidenciando, no contexto da detecção de anomalias, uma menor probabilidade de ser considerado ataque. Pontos com valores positivos no eixo x demonstram a tendência do modelo de considerar a amostra mais anômala. Saídas do modelo que ultrapassam um limiar pré-definido são consideradas ataques pela 1D-CNN-GAN.

Sistemas de detecção de intrusão baseados em anomalias são construídos sob a premissa de que o tráfego de um ataque destoa do comportamento legítimo dos fluxos IP da rede. Valores de SHAP para características associadas ao tráfego normal concentram-se próximos do valor esperado, refletindo os padrões aprendidos pelo modelo e menor variação em relação ao SHAP zero. A Figura 12a mostra o gráfico de SHAP para o tráfego legítimo do dia de teste do CIC-DDoS2019, no qual a maioria das características apresentou clara separação por valor.

As três variáveis mais impactantes da Figura 12a, que incluem as entropias de endereços IP de origem e destino em diferentes intervalos, não se concentram próximas ao valor esperado. Esse comportamento pode ser relacionado a falsos positivos, dado que os valores menores dessas variáveis impactam positivamente a saída do modelo. A Figura 10, da análise exploratória de dados, reforça essa hipótese, exibindo a maior concentração de ataques em intervalos menores das entropias de endereço IP de origem e destino.

As Figuras 12b, 12c, 12d e 12e exibem os gráficos de enxame de SHAP dos ataques NetBIOS, LDAP, MSSQL e UDP, respectivamente. Essas anomalias podem ser detectadas visto que afetam a saída do modelo ao aumentar a probabilidade de tráfego ser uma intrusão. O SHAP de todos os ataques mostrou uma tendência positiva carregada pela entropia de porta de destino (*i.e.*, H(Porta dest)). As demais características causam influência negligenciável, tanto positiva quanto negativa. A entropia de porta de destino no quarto carimbo temporal, denotada como H(Porta dest)_t-4, tem o impacto negativo mais acentuado, impactando a decisão do modelo para inferências mais incertas.

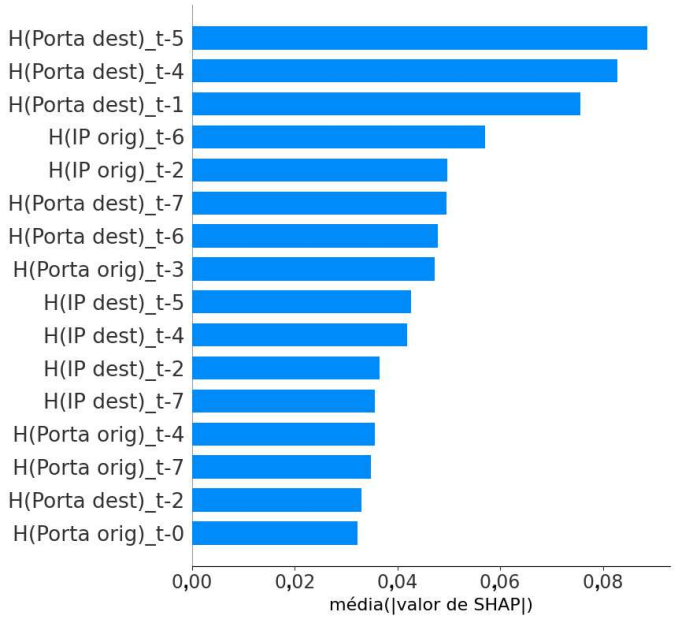

Figura 13 – Impacto médio na saída do modelo para o dia de teste do conjunto de dados CIC-DDoS2019 antes da otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

A Figura 13 apresenta o valor médio absoluto de SHAP para cada característica do conjunto de teste. Esse gráfico desconsidera a orientação dos impactos das variáveis, revelando apenas a magnitude das influências. A entropia de porta de destino nos instantes 5, 4 e 1 exibe o maior impacto absoluto, alinhando-se com a análise do tráfego separado por categoria exposta pela Figura 12.

As Figuras 14 e 15 são gráficos de mapa de calor de SHAP. Esses expõem o comportamento temporal do modelo para os ataques MSSQL e UDP, respectivamente. O eixo x demarca o número da instância ordenada cronologicamente do início ao fim do dia de teste. O eixo y indica os nomes das características. A função $f(x)$ presente na parte superior do gráfico expõe o desvio da saída do modelo em relação ao valor esperado, indicado pela linha horizontal tracejada. As barras horizontais à direita da figura representam o valor médio absoluto dos valores de SHAP para cada característica. Cada instância pode ser analisada verticalmente, tendo as características correspondentes coloridas de acordo com sua influência na decisão do modelo. Amostras marcadas em azul indicam influências negativas, já as vermelhas, positivas.

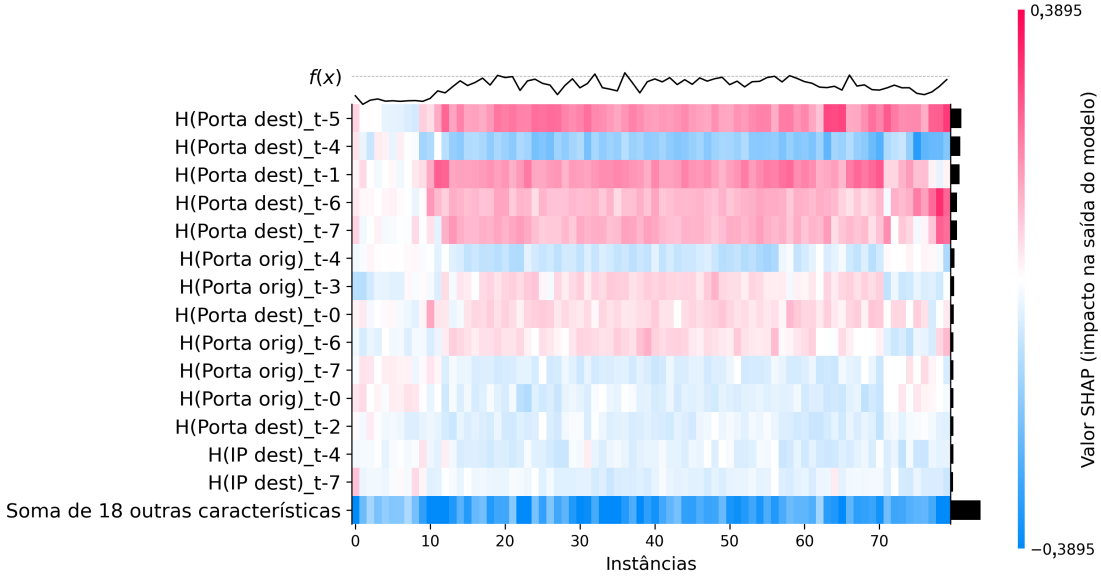

Figura 14 – Impactos temporais SHAP das características para o ataque MSSQL do conjunto de dados CIC-DDoS2019 antes da otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

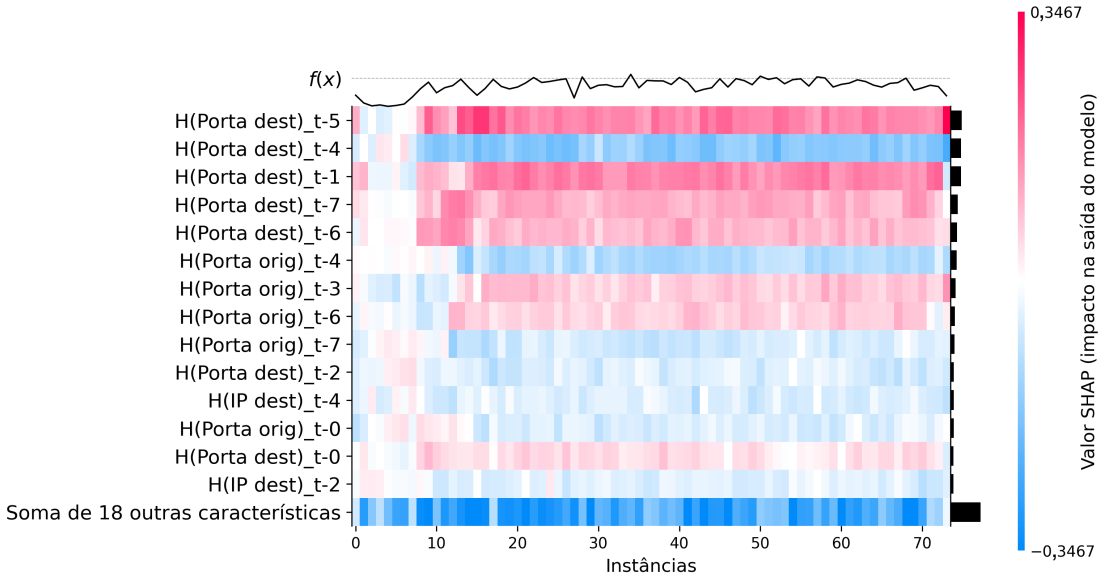

Figura 15 – Impactos temporais SHAP das características para o ataque UDP do conjunto de dados CIC-DDoS2019 antes da otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

Esses ataques foram escolhidos por apresentarem um padrão de menor impacto no primeiro grupo de instâncias, comportamento ausente nas demais categorias de tráfego. Esse padrão pode estar relacionado à estratégia de janela deslizante do modelo original, cujo tamanho é de oito segundos. Esse valor coincide aproximadamente com o ponto em que o ataque começa a refletir aumento no impacto das características, especialmente para o ataque de negação de serviço distribuído UDP. Esse alinhamento poderia implicar a presença do efeito de atraso temporal introduzido pela estratégia de janela deslizante, mas provavelmente está ligado à distribuição dos dados. No conjunto de dados CIC-DDoS2019, os ataques têm início e fim graduais. Essa hipótese pode ser reforçada ao observar o comportamento mais gradual dessas classes de dados em relação às demais na Figura 8. A parcela final desses ataques não apresentou mudança no impacto das características.

5.5 Atualização do Sistema

Dadas as explicações a respeito da influência das características providas pela análise inicial do SHAP, conclui-se que a entropia de porta de destino é a mais impactante na tarefa de detectar ataques. Portanto, o sistema de detecção foi reconfigurado como descrito no artigo original por Ruffo *et al.* [98], porém, incluindo apenas a entropia de porta de destino como entrada de dados. Essa redução de dimensionalidade reflete a descoberta do SHAP de que as demais características introduziam ruído preditivo ou redundância, prejudicando a capacidade de generalização do discriminador. A otimização culminou, ainda, em mudanças na arquitetura do modelo: houve a compressão da dimensão do

espaço latente de 64 para 32; a redução dos neurônios da rede geradora, de 32 para 16; o acréscimo de oito neurônios na rede discriminadora; e a expansão da janela deslizante de 8 para 16 segundos.

A Tabela 5 compara as métricas de desempenho do modelo original com o otimizado para o dia de teste do CIC-DDoS2019. As métricas mostram uma melhora nos resultados, com um avanço absoluto de 0,047 no MCC, de 0,804 para 0,851, e de 3,6 p.p. no F1-Score, de 0,865 para 0,901, o que corresponde a ganhos relativos de 5,8% e 4,2%, respectivamente. Para este *dataset*, o número de ocorrências de falsos negativos reduziu-se de 693 para 509, uma queda de 26,6%, refletindo um avanço de 6,0 p.p. na métrica de revocação, que passou de 0,773 para 0,833.

Tabela 5 – Comparação de desempenho entre o NIDS baseado em 1D-CNN-GAN original e o otimizado para o dia de teste do conjunto de dados CIC-DDoS2019. **Fonte:** elaboração do próprio autor.

Métrica	1D-CNN-GAN original	1D-CNN-GAN otimizado
F1-Score	0,865	0,901
MCC	0,804	0,851
Precisão	0,981	0,981
Revocação	0,773	0,833
Tempo de inferência (segundos)	2,525	3,192

O tempo de inferência para o conjunto de teste apresentou um incremento de aproximadamente 0,67 segundos em decorrência da nova configuração de hiperparâmetros. Esse aumento traduz-se em uma latência adicional de aproximadamente 88 μs na predição de cada janela temporal de um segundo, um valor potencialmente negligenciável. Embora o novo sistema seja marginalmente mais lento na etapa de classificação, sua arquitetura opera sobre uma entrada de dados de menor dimensionalidade. Essa redução alivia a carga computacional exigida do mecanismo de extração de características de fluxos IP, resultando em um sistema de segurança menos propenso a gargalos operacionais e mais viável para implantação em ambientes de produção reais.

5.6 Explicação Final

O gráfico de SHAP para o modelo 1D-CNN-GAN otimizado, ilustrado na Figura 16, revela um padrão de espalhamento similar ao do modelo original. Isso se dá pelo fato de as características mais importantes do gráfico original (diferentes instantes da entropia de porta de destino) terem sido mantidas para o modelo otimizado. O sexto carimbo temporal dessa característica aparece como o mais influente na Figura 17. De modo geral, as entropias de porta de destino se condensam em torno do valor zero de SHAP para o tráfego normal e apresentam impacto positivo na saída do modelo para os ataques considerados. Esse comportamento é desejado, uma vez que ele guia o modelo para inferências corretas.

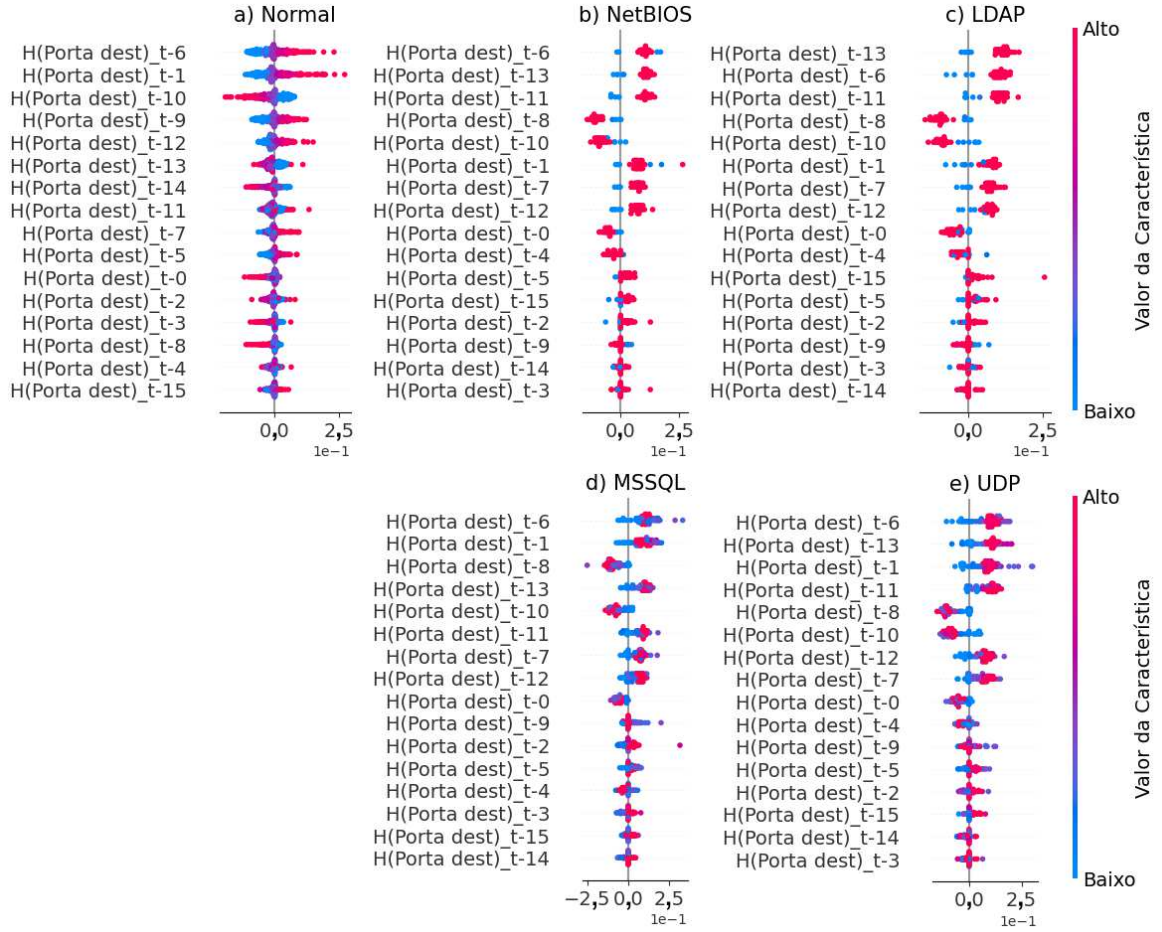

Figura 16 – Gráfico de enxame do dia de teste do conjunto de dados CIC-DDoS2019 separado por categoria após otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

Em contrapartida, os carimbos temporais 8 e 10 consistentemente apresentam efeitos negativos na saída do modelo para todos os tipos de tráfego, indicando a necessidade de um processo mais refinado de otimização de hiperparâmetros. Embora as características prejudiciais ao modelo estejam presentes na versão otimizada do sistema, suas magnitudes

de impacto são aproximadamente 20 a 25% menos significativas que as da característica mais importante. Essa magnitude é exibida na Figura 17, evidenciada pelo contraste entre as barras das variáveis H(Porta dest)_t-10 e H(Porta dest)_t-8 comparadas com H(Porta dest)_t-6.

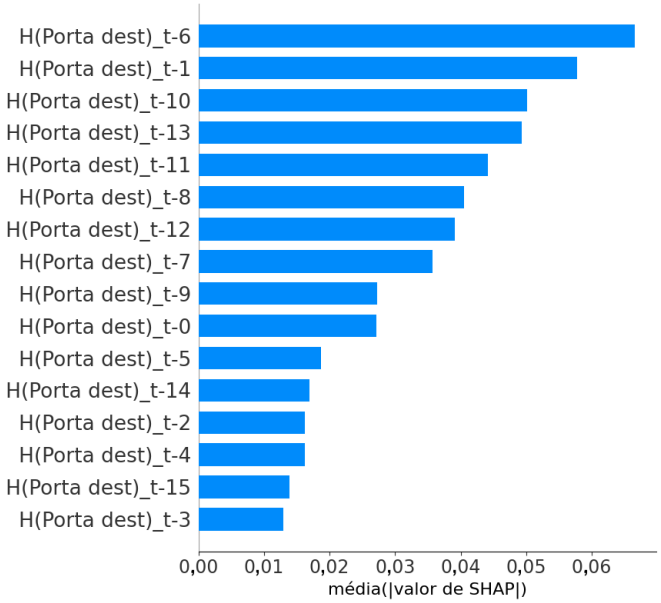

Figura 17 – Impacto médio na saída do modelo para o dia de teste do conjunto de dados CIC-DDoS2019 após otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

As Figuras 18 e 19 retratam o comportamento temporal do modelo otimizado para os ataques MSSQL e UDP. Os mesmos ataques foram escolhidos posto que eles diferem das categorias de tráfego remanescentes, similarmente ao modelo original. No entanto, no modelo otimizado, os ataques apresentam uma diminuição na importância das características próximas ao final do recorte temporal analisado. Decaimento de influência pode ser descartado, já que $f(x)$ não tem um declínio consistente ao longo do tempo. Ambos os ataques exibem oscilação em seu final, particularmente a negação de serviço distribuído UDP. O início e o término graduais dos ataques representam a explicação mais plausível para esse comportamento.

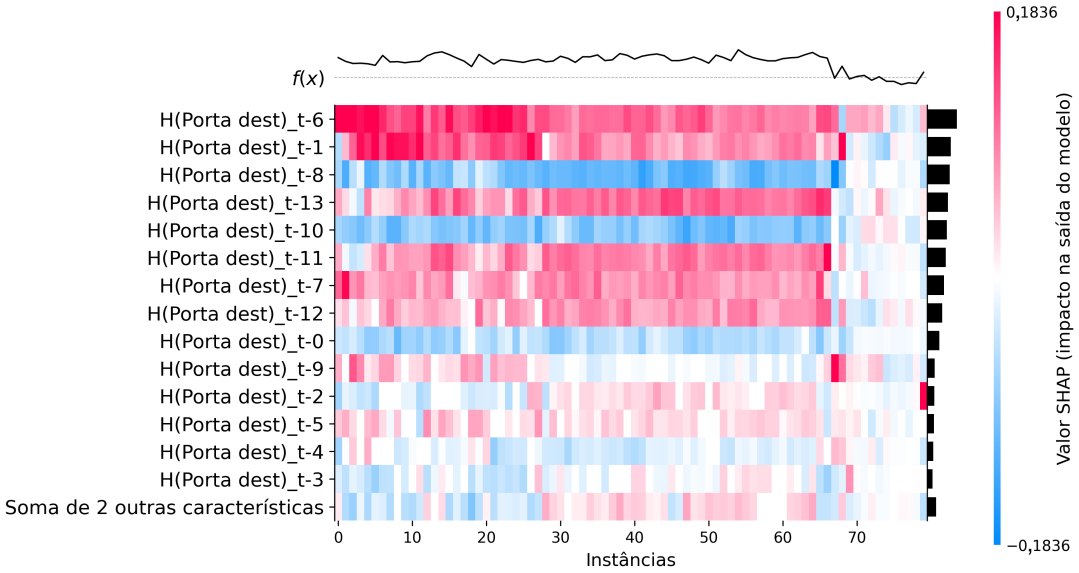

Figura 18 – Impactos temporais SHAP das características para o ataque MSSQL do conjunto de dados CIC-DDoS2019 após otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

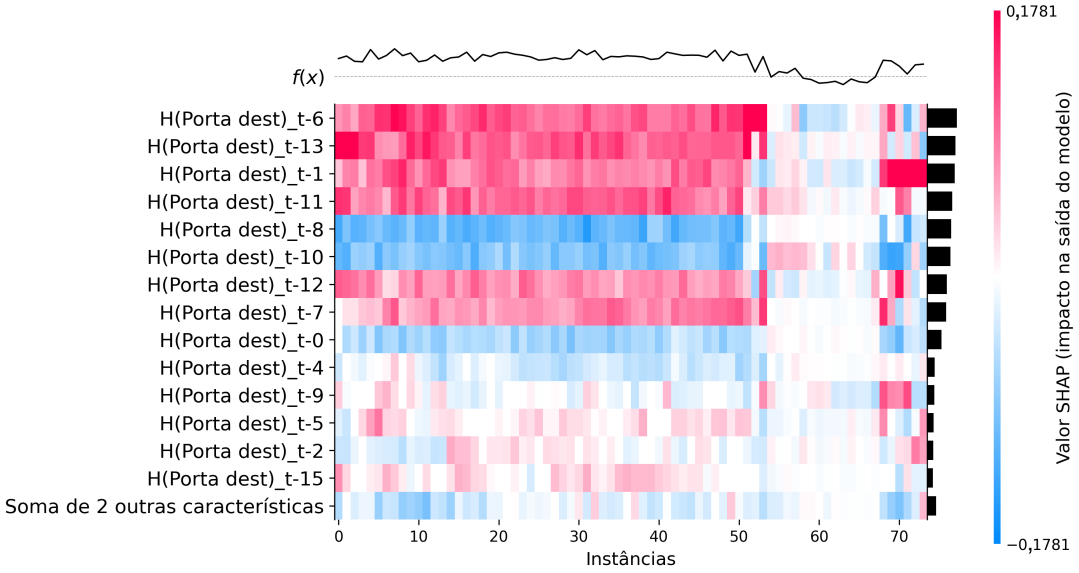

Figura 19 – Impactos temporais SHAP das características para o ataque UDP do conjunto de dados CIC-DDoS2019 após otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

5.7 Validação da Metodologia

Para atestar a capacidade de generalização da metodologia proposta, esta seção estende os experimentos para novos cenários em duas frentes. Primeiro, avalia-se o modelo principal, a arquitetura 1D-CNN-GAN, no conjunto de dados CSE-CIC-IDS2018, ambiente não explorado em seu trabalho original. Em seguida, demonstra-se a independência arquitetural do método ao aplicar o mesmo processo de otimização a um modelo alterna-

tivo baseado em β-VAE sobre três bases distintas: CIC-DDoS2019, CSE-CIC-IDS2018 e Orion2026.

5.7.1 1D-CNN-GAN no cenário CSE-CIC-IDS2018

O modelo 1D-CNN-GAN com o conjunto de características original de quatro entropias produziu um MCC de 0,703, F1-Score de 0,726, Precisão de 0,827 e Revocação de 0,647 para o conjunto de dados CSE-CIC-IDS2018. Essas métricas demonstram a presença proeminente de falsos negativos nas predições do modelo. A Figura 20a retrata o gráfico de enxame das explicações do modelo aplicado ao tráfego legítimo, enquanto a Figura 20b apresenta as explicações para o DDoS-HOIC.

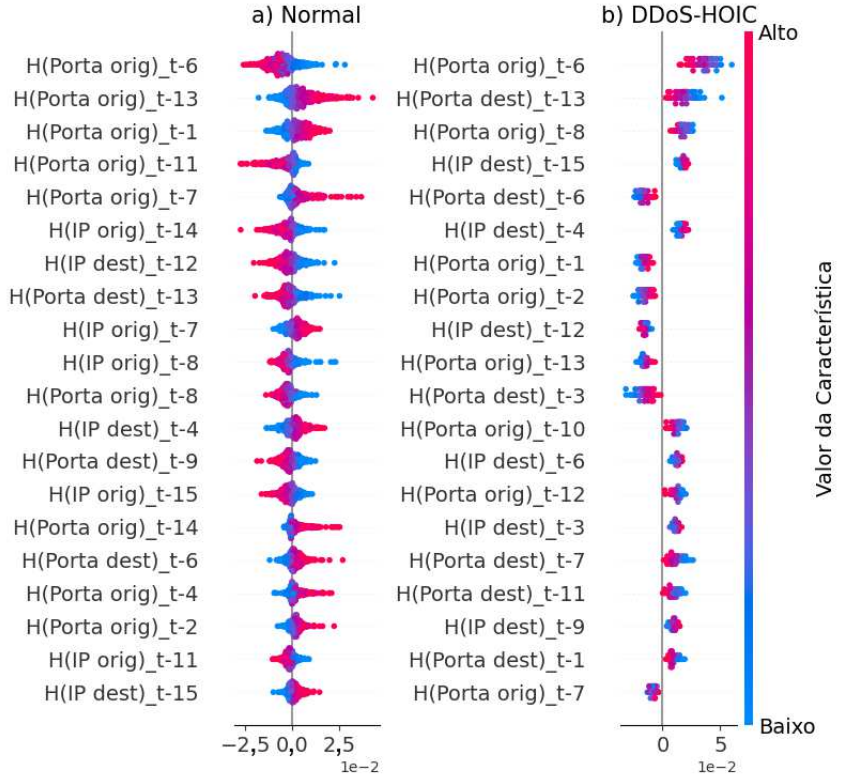

Figura 20 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 antes da otimização do NIDS baseado em 1D-CNN-GAN separado por categorias de tráfego. **Fonte:** elaboração do próprio autor.

A Figura 20a sugere que a entropia de porta de origem pode ser relacionada a falsos positivos, já que essa característica é a que mais desvia do valor esperado para valores maiores de SHAP. A entropia de porta de destino para tráfego legítimo não desviou significativamente do valor esperado. Na Figura 20b, a entropia de endereço IP de origem não apareceu entre as 20 características mais importantes, mostrando baixa relação com a classificação do modelo. As demais características apresentaram um padrão de espalhamento em valores de SHAP positivos e negativos para carimbos temporais distintos. A entropia de porta de destino é a segunda mais impactante e a mais estável de maneira

geral, mostrando predominantemente influência positiva para diferentes instantes desse ataque.

A Figura 21 mostra o comportamento temporal do sistema sob o escopo do SHAP para o ataque DDoS-HOIC. Ao longo do dia, $f(x)$ oscila próximo ao valor esperado, gerando falsos negativos. Apesar de $f(x)$ flutuar, não há evidência visual suficiente para relacionar esse comportamento com nenhuma característica específica. Portanto, nenhum atributo pode ser considerado errôneo e responsável por essa variação.

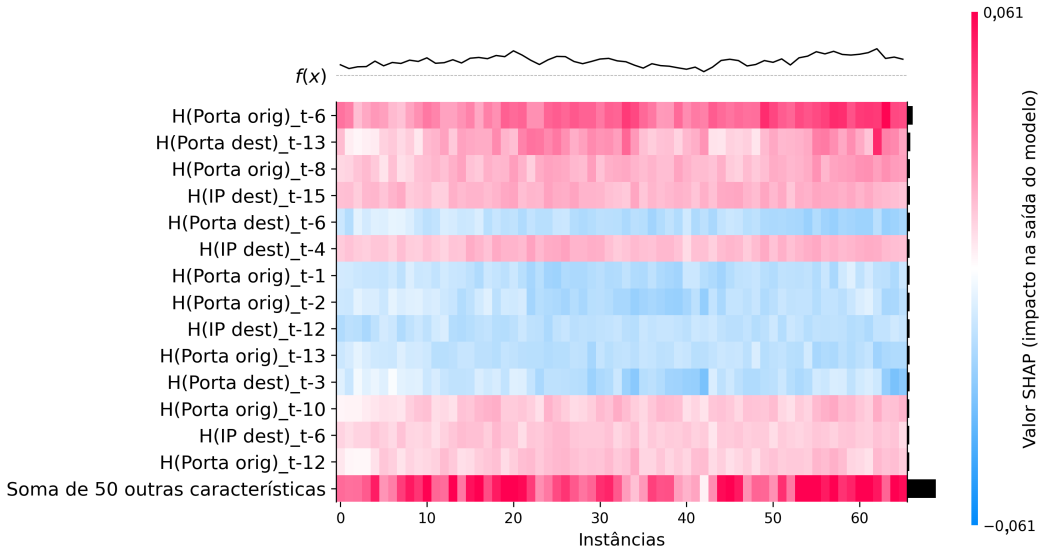

Figura 21 – Impactos temporais SHAP das características para o ataque DDoS-HOIC do conjunto de dados CSE-CIC-IDS2018 antes da otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

Similar ao experimento conduzido no *dataset* CIC-DDoS2019, a entropia de porta de destino se destaca como a mais influente na identificação de comportamento volumétrico atípico. Por isso, o modelo foi retreinado usando apenas essa característica e atingiu 0,916 para a métrica MCC, 0,925 para F1-Score, 0,881 para Precisão e 0,972 para Revocação. O modelo diminuiu a quantidade de falsos negativos de 1.159 para 89 e de falsos positivos de 445 para 428. Esses resultados refletem um avanço absoluto de 0,213 na métrica MCC, de 0,703 para 0,916, o que equivale a um ganho relativo de 30,3%, e uma diminuição de 92,3% no número de falsos negativos e de 3,8% no de falsos positivos. O tempo de inferência para o dia de teste completo diminuiu em 0,05 segundos, o que indica empate para essa métrica.

A Figura 22 mostra o gráfico de enxame das explicações de SHAP para o modelo otimizado. Carimbos temporais mais altos, *i.e.*, instantes mais distantes do corrente, como as entropias de porta de destino nos segundos 10, 11 e 13, exibem maior inconsistência. Especificamente, os carimbos temporais 11 e 13 representam alguns dos maiores desvios do valor esperado para o tráfego normal, enquanto os segundos 10 e 11 consisten-

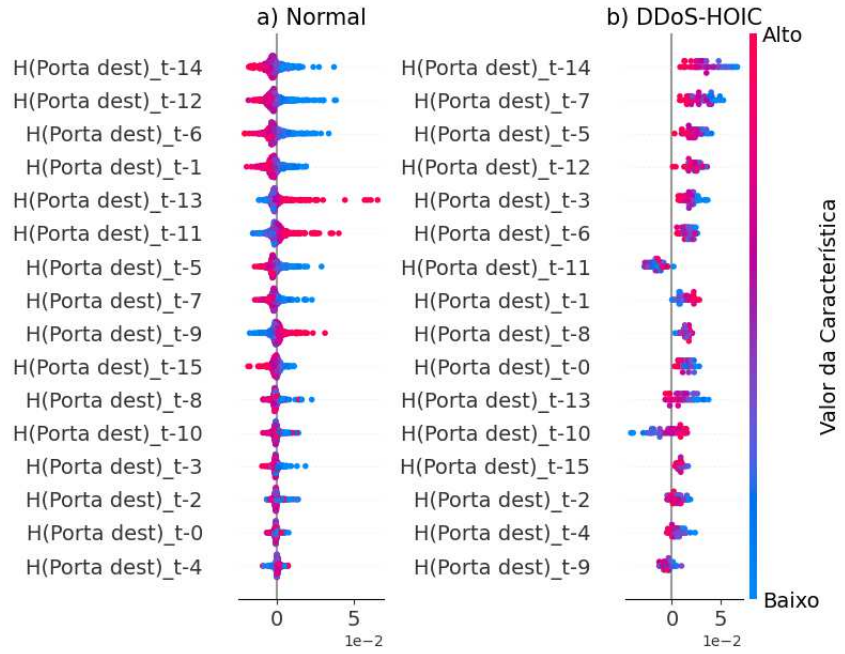

Figura 22 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 após otimização do NIDS baseado em 1D-CNN-GAN separado por categorias de tráfego. **Fonte:** elaboração do próprio autor.

temente dirigiram as predições no sentido de valores mais baixos de SHAP. O segundo 13 apresentou comportamento mais próximo ao esperado para o tráfego de ataque. Esse padrão comportamental sugere a necessidade de implementar uma técnica mais refinada de otimização de hiperparâmetros.

A Figura 23 mostra o SHAP temporal para o tráfego de ataque DDoS-HOIC, evidenciando menos falsos negativos visto que $f(x)$ desvia mais do valor esperado comparado com o modelo original. Como apresentado no gráfico de enxame, os carimbos temporais 10 e 11 possuem impacto menor, possivelmente causando flutuação na saída do modelo para esse ataque. Além disso, nem o efeito de atraso temporal, nem o decaimento de influência foram observados tanto no modelo original quanto no otimizado.

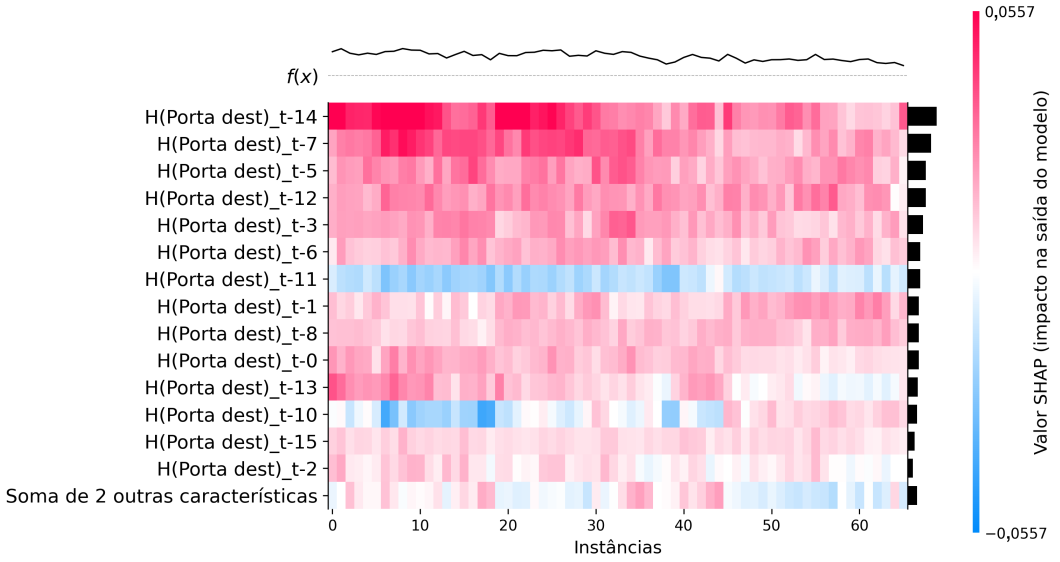

Figura 23 – Impactos temporais SHAP das características para o ataque DDoS-HOIC do conjunto de dados CSE-CIC-IDS2018 após otimização do NIDS baseado em 1D-CNN-GAN. **Fonte:** elaboração do próprio autor.

A versão otimizada do sistema de detecção de intrusão apresentada neste trabalho é especializada em detectar ataques volumétricos como Negação de Serviço. Esses ataques perturbam a distribuição normal de tráfego que flui para a vítima a partir de atacantes. A redução do conjunto de características para apenas a entropia de porta de destino preserva a estratégia de perturbação estatística para esse tipo de ataque. No entanto, desvio de conceito (*concept drift*) pode comprometer a capacidade do sistema de modelar o tráfego normal, impossibilitando a definição de uma linha base e a detecção de desvios causados por ataques. Para abordar esse problema, o sistema de detecção de intrusão deve ser resiliente a mudanças na distribuição dos dados.

5.7.2 β-VAE no cenário CIC-DDoS2019

Testando o IDS baseado em β-VAE no cenário de dados CIC-DDoS2019, o sistema obteve MCC de $0,729$, F1-Score de $0,721$, Precisão de $0,591$ e Revocação de $0,925$. A taxa de falsos positivos, de $0,024$, mostrou-se a principal causa de redução da assertividade do sistema. A Figura 24 ilustra o gráfico de enxame das explicações do modelo neste conjunto de dados para as cinco categorias de tráfego separadamente.

A Figura 24a sugere baixa estabilidade do sistema na modelagem do tráfego legítimo, exibindo espalhamento das características em torno do valor esperado. As entropias de porta, para esta categoria de tráfego, concentram-se mais em torno do valor esperado. Para os quatro ataques distintos, da Figura 24b a 24e, a entropia de porta de destino possui o maior impacto positivo na saída do modelo, especialmente para valores mais altos desse atributo. As demais características concentram-se próximas ao valor esperado.

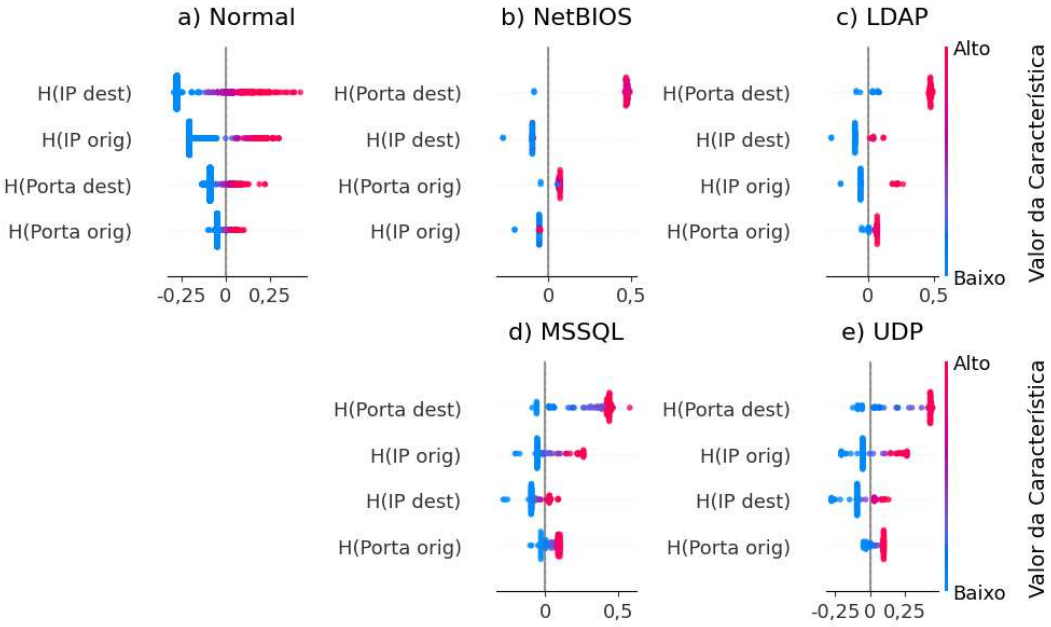

Figura 24 – Gráfico de enxame do dia de teste do conjunto de dados CIC-DDoS2019 separado por categoria antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

O comportamento temporal do sistema baseado em β-VAE assemelha-se ao 1D-CNN-GAN, em que os ataques DDoS UDP (Figura 25) e MSSQL (Figura 26) apresentam valores de SHAP menores para amostras ao fim do conjunto de teste. Essa evidência é insuficiente para apontar decaimento de influência, mas mostra que a entropia de IP de origem compensa essa queda ao aumentar a saída do modelo para as amostras ao final do conjunto.

Tendo em vista o comportamento consistente das entropias de aproximarem-se do valor esperado, como exibido no gráfico de enxame, optou-se por manter apenas a característica de entropia de porta de destino. Essa característica demonstrou-se a mais estável dentre as situações analisadas, constantemente aumentando a pontuação de anomalia do modelo para amostras espúrias e concentrando-se no valor esperado para amostras legítimas.

Quando retreinado utilizando exclusivamente a entropia da porta de destino, o sistema alcançou um MCC de $0,845$, F1-Score de $0,846$, precisão de $0,935$ e revocação de $0,773$, passando dos $0,27$ segundos originais na detecção de todo o conjunto de teste para $0,21$ segundos. A taxa de falsos positivos reduziu para $0,001$, reflexo da queda no número de ocorrências legítimas incorretamente classificadas como espúrias, que passou de 1.966 para apenas 165. Embora a taxa de falsos negativos passe de 7,4% para 22,7%, um acréscimo de 15,3 p.p., esse *trade-off* com menor punição de agentes legítimos é considerado desejável sob a perspectiva da gerência de redes de computadores, visto que mais de 77% dos ataques totais ainda são identificados. Ademais, a modelagem baseada

em entropia captura prioritariamente os ataques de maior impacto volumétrico. Mesmo que uma parcela minoritária do tráfego espúrio não seja bloqueada, o sistema ainda é capaz de diminuir a eficácia do ataque em congestionar suas vítimas.

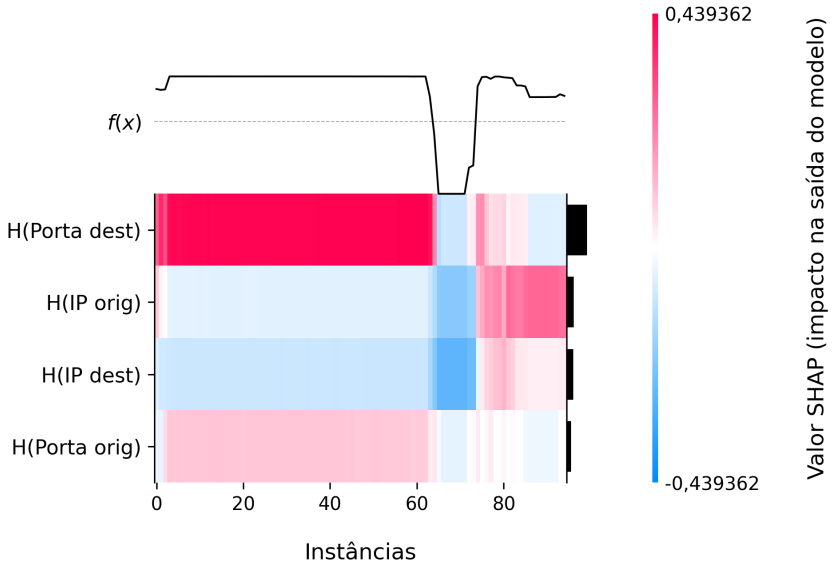

Figura 25 – Impactos temporais SHAP das características para o ataque UDP do conjunto de dados CIC-DDoS2019 antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

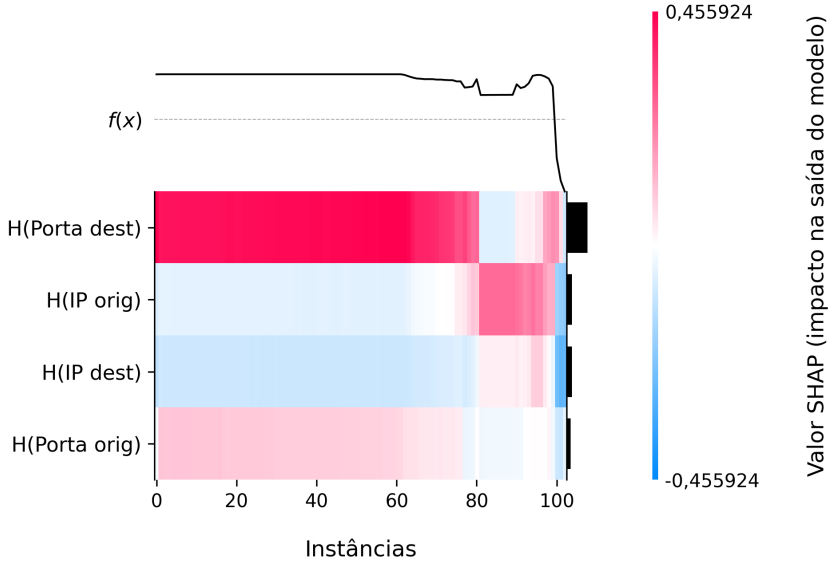

Figura 26 – Impactos temporais SHAP das características para o ataque MSSQL do conjunto de dados CIC-DDoS2019 antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

O comportamento do modelo atualizado é demonstrado na Figura 27. O tráfego legítimo, detalhado na Figura 27a, apresenta uma maior concentração de amostras em valores de SHAP inferiores ao valor esperado. Como o sistema é treinado exclusivamente com dados legítimos, a modelagem do valor esperado denota a ocorrência de tráfego normal; logo, valores abaixo desse limiar indicam uma probabilidade de ataque ainda menor. Para o modelo otimizado, essa concentração de amostras ocorre em valores de SHAP duas vezes menores do que os observados no modelo original, o que demonstra maior estabilidade na modelagem dos dados legítimos. Quanto aos ataques, ilustrados nas Figuras 27b a 27e, constata-se uma melhor separação dos valores de SHAP no modelo otimizado em comparação ao original, vista a menor disposição de instâncias sobre o valor esperado. Os falsos negativos ainda presentes são justificados pela concentração de amostras espúrias em valores de SHAP negativos, o que indica baixo volume de tráfego para os ataques em determinado período de sua execução.

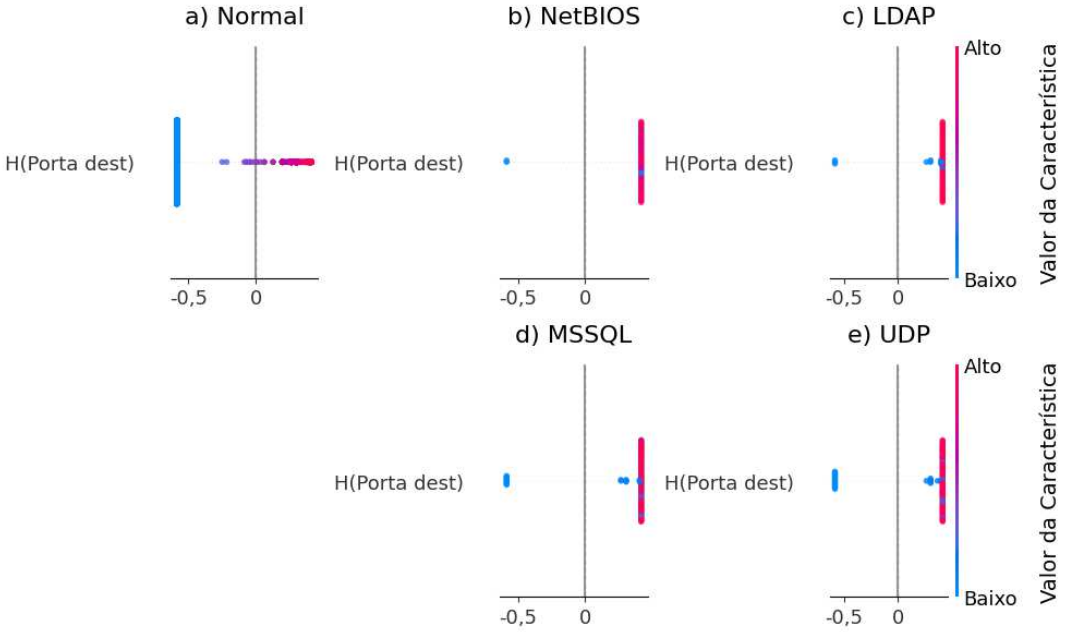

Figura 27 – Gráfico de enxame do dia de teste do conjunto de dados CIC-DDoS2019 separado por categoria depois da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Do ponto de vista temporal, o perfil de SHAP do sistema otimizado baseado em β-VAE reforça a análise do baixo volume de tráfego durante o ataque, mantendo o comportamento observado no modelo baseado em 1D-CNN-GAN. Constata-se uma inversão na ocorrência de falsos negativos do final da janela de amostras para o início. Esse fenômeno enfraquece a hipótese de decaimento de influência ou de atraso na detecção, demonstrando o início e término graduais desses ataques. Esse comportamento temporal pode ser visualizado na Figura 28, que detalha temporalmente as ocorrências dos ataques UDP (Figura 28a) e MSSQL (Figura 28b).

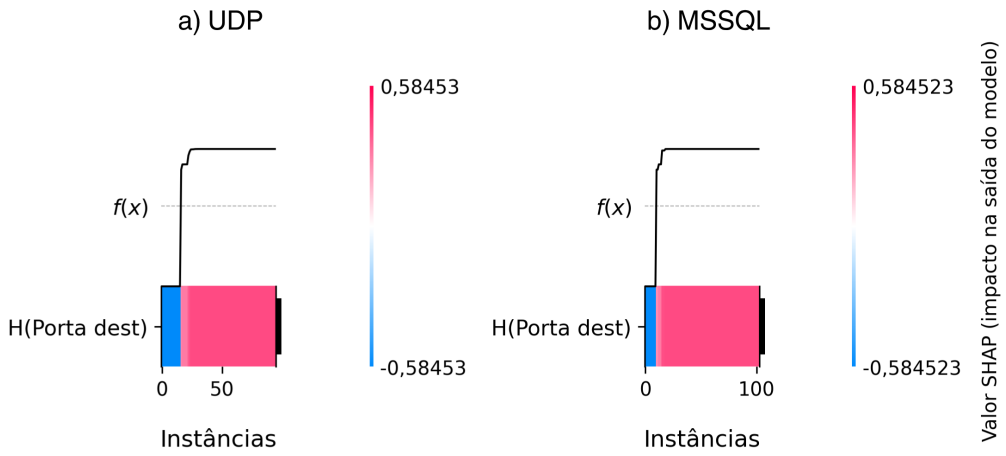

Figura 28 – Impactos temporais SHAP da característica para os ataques UDP e MSSQL do conjunto de dados CIC-DDoS2019 depois da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

5.7.3 β-VAE no cenário CSE-CIC-IDS2018

O sistema baseado em β-VAE apresentou, originalmente, um MCC de $0,786$, F1-Score de $0,855$, precisão de $0,987$, revocação de $0,753$ e $0,29$ segundos para discriminar todo o conjunto de teste. A taxa de falsos negativos demonstrou-se a maior limitação da modelagem, atingindo um valor de $0,246$, o que se reflete na diminuição da métrica de revocação. Esse comportamento evidencia-se pela pronunciada influência da entropia de IP de destino no ataque DDoS-HOIC, ilustrado na Figura 29b, concentrando-se em valores de SHAP negativos. Para o tráfego legítimo, essa característica foi a que mais desviou do valor esperado, enquanto as demais ficaram mais próximas dele.

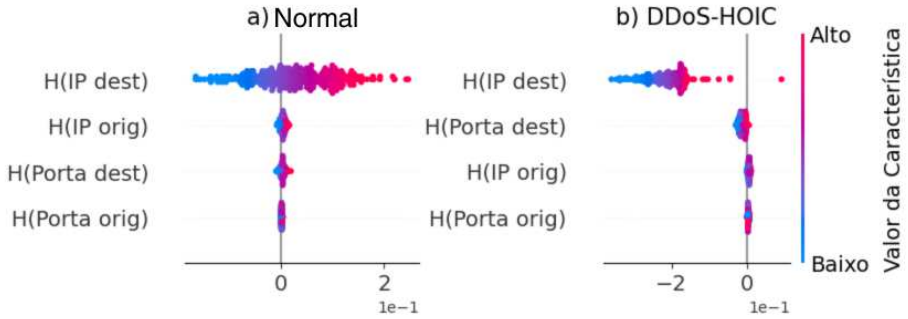

Figura 29 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 separado por categoria antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Vista a falta de discernimento acerca da influência das demais características, removeu-se apenas a entropia de IP de destino e retreinou-se o sistema. A taxa de falsos positivos manteve-se estável, enquanto a taxa de falsos negativos recuou 9,7 p.p., caindo

de 0, 246 para 0, 149. Essa diminuição resultou em um MCC de 0, 862, F1-Score de 0, 913, precisão de 0, 987, revocação de 0, 850 e detecção do conjunto de teste em 0, 17 segundos. O comportamento desta rede é ilustrado na Figura 30, na qual é possível detectar a mesma tendência do modelo antes da otimização. A entropia de IP de origem domina a influência do modelo, causando falsos negativos.

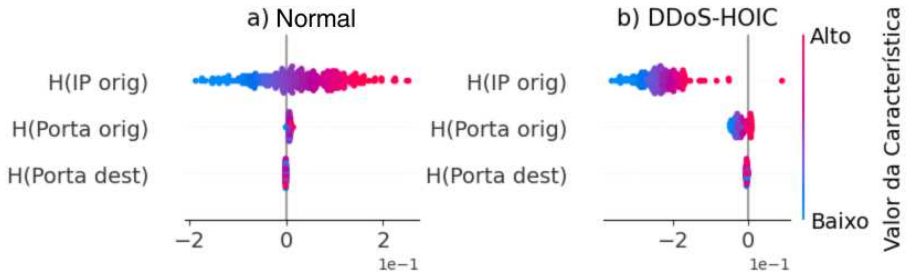

Figura 30 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 separado por categoria depois da primeira rodada de otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Seguindo a metodologia cíclica proposta, uma nova rodada de otimização é executada removendo a entropia de IP de origem. A modelagem resultante atingiu MCC de 0, 843, F1-Score de 0, 898, precisão de 0, 992, revocação de 0, 821 e 0, 17 segundos para detecção completa. O aumento da métrica de precisão e a diminuição da revocação revelam que essa versão do sistema reduziu a quantidade de falsos positivos, enquanto aumentou falsos negativos. A explicação para esse fenômeno é evidenciada na Figura 31. A entropia de porta de origem reduziu o espalhamento para SHAP negativo no tráfego legítimo (Figura 31a) em relação à versão anterior, enquanto manteve o espalhamento positivo. Para o ataque, essa característica apresentou tendência em causar falsos negativos, concentrando-se em SHAP negativo (Figura 31b).

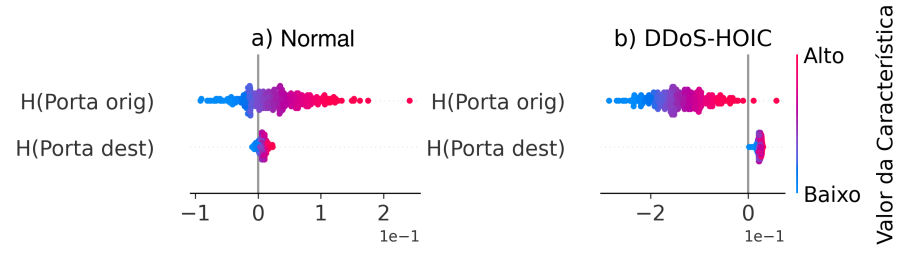

Figura 31 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 separado por categoria depois da segunda rodada de otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Finalmente, a característica responsável pelo comportamento indesejado é removida, e o sistema é retreinado utilizando apenas a variável remanescente. Baseando-se exclusivamente na entropia da porta de destino, o modelo obteve um MCC de $0,940$, F1-Score de $0,966$, precisão de $0,969$, revocação de $0,962$ e detecção em $0,17$ segundos. Essa configuração apresentou o melhor equilíbrio entre as taxas de falsos negativos e positivos, as quais assumiram os valores de $0,037$ e $0,023$, respectivamente.

A dinâmica desse modelo é explicada na Figura 32. Para o tráfego legítimo (Figura 32a), observa-se uma tendência da característica mantida em induzir falsos positivos, devido à sua disposição em valores de SHAP superiores ao esperado. Esse padrão é compensado na modelagem por valores de SHAP ainda maiores atribuídos ao tráfego espúrio (Figura 32b). Esse comportamento indica que os dados de teste são, de modo geral, mais volumosos que os do dia de treino, o que justifica o deslocamento de todo o tráfego para patamares mais elevados de SHAP. Assim, a capacidade do modelo de discriminar as anomalias do comportamento esperado decorre do incremento proporcional na previsão do tráfego espúrio em relação ao legítimo.

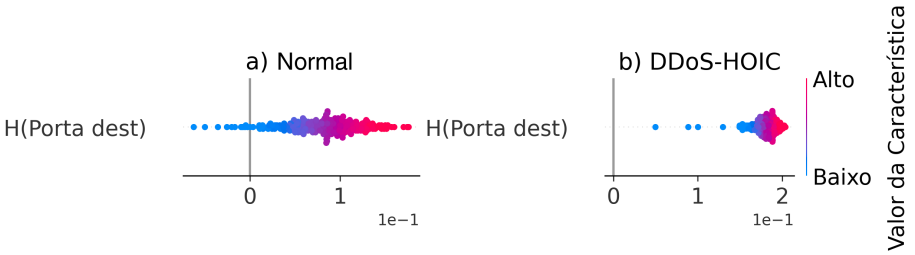

Figura 32 – Gráfico de enxame do dia de teste do conjunto de dados CSE-CIC-IDS2018 separado por categoria depois da terceira rodada de otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Ao longo das rodadas de otimização, o comportamento temporal do sistema refletiu a dinâmica do tráfego. A Figura 33a retrata o modelo original com quatro variáveis de entropia, no qual se observa o decaimento de $f(x)$ aliado à redução do SHAP da característica dominante. Após a primeira otimização (Figura 33b), $f(x)$ cresce impulsionado pelo SHAP da nova entropia dominante, tendência que se repete no sistema com duas variáveis (Figura 33c). Esse panorama se consolida na Figura 33d, na qual $f(x)$ se posiciona acima do valor esperado, sinalizando a detecção de verdadeiros positivos. Nos instantes finais, a redução de $f(x)$ sugere diminuição no volume do ataque em seu encerramento. Descarta-se decaimento de influência e reforça-se a hipótese de atenuação final do ataque, vista a redução concomitante no valor de SHAP da característica, como observado na ampliação dos instantes finais do ataque. Esse padrão indica que o modelo responde à perda de intensidade do surto volumétrico espúrio.

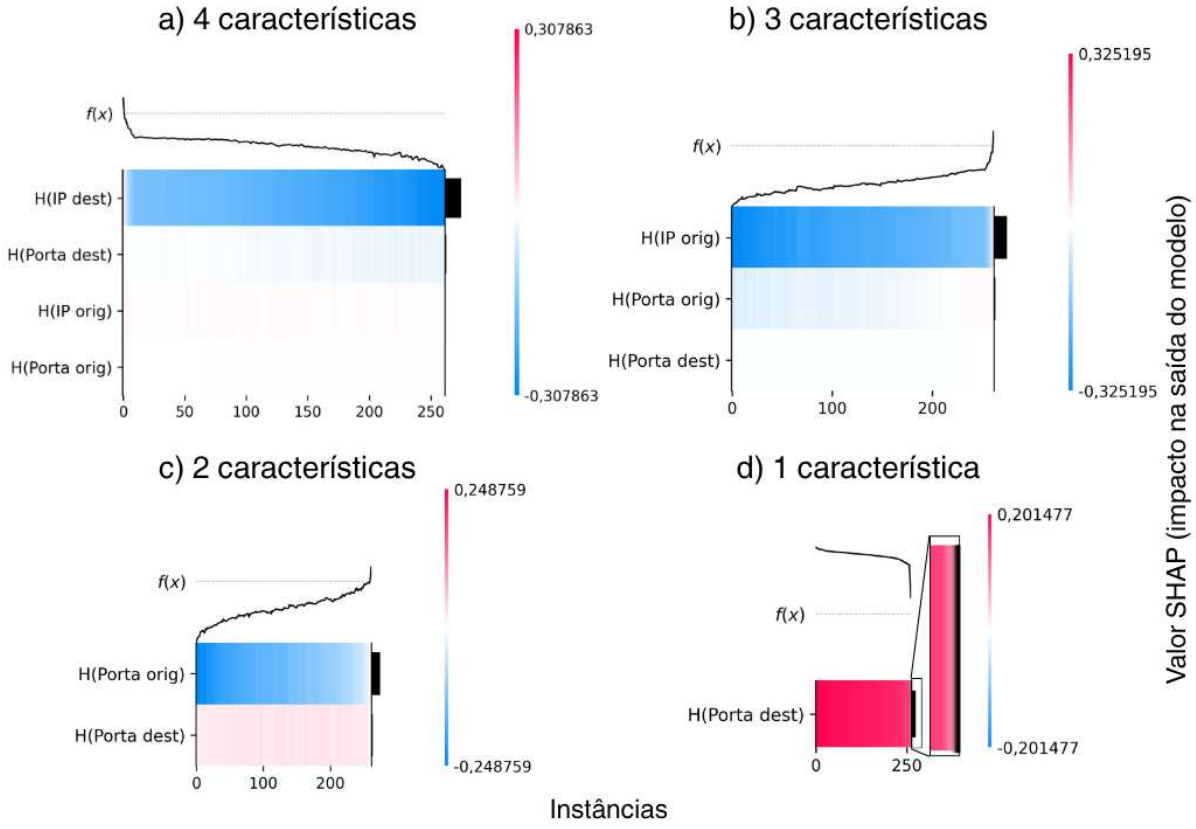

Figura 33 – Impactos temporais SHAP das características para o ataque DDoS-HOIC do conjunto de dados CSE-CIC-IDS2018 nas configurações exploradas do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

5.7.4 β-VAE no cenário Orion2026

As métricas de desempenho do sistema baseado em β-VAE para as quatro características originais de entropia são semelhantes entre as diferentes cardinalidades de ataque presentes nos dias de teste do conjunto Orion2026. A maior oscilação ocorre na precisão, que varia de $0,906$ a $0,898$ do terceiro ao quinto dia. Embora as métricas globais indiquem uma taxa de falsos positivos superior à de falsos negativos, o mapeamento via SHAP expõe uma modelagem estável para o tráfego legítimo, porém com tendências de indução de falsos negativos. O comportamento do sistema para o terceiro, quarto e quinto dias é ilustrado nas Figuras 34, 35 e 36, respectivamente.

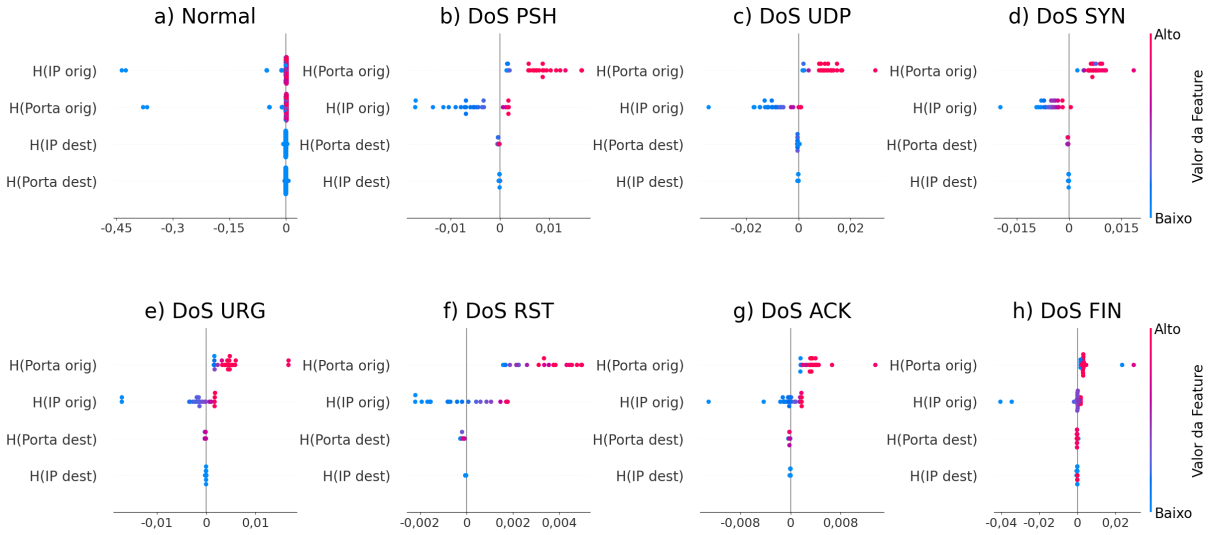

Figura 34 – Gráfico de enxame do terceiro dia do conjunto de dados Orion2026 separado por categoria antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

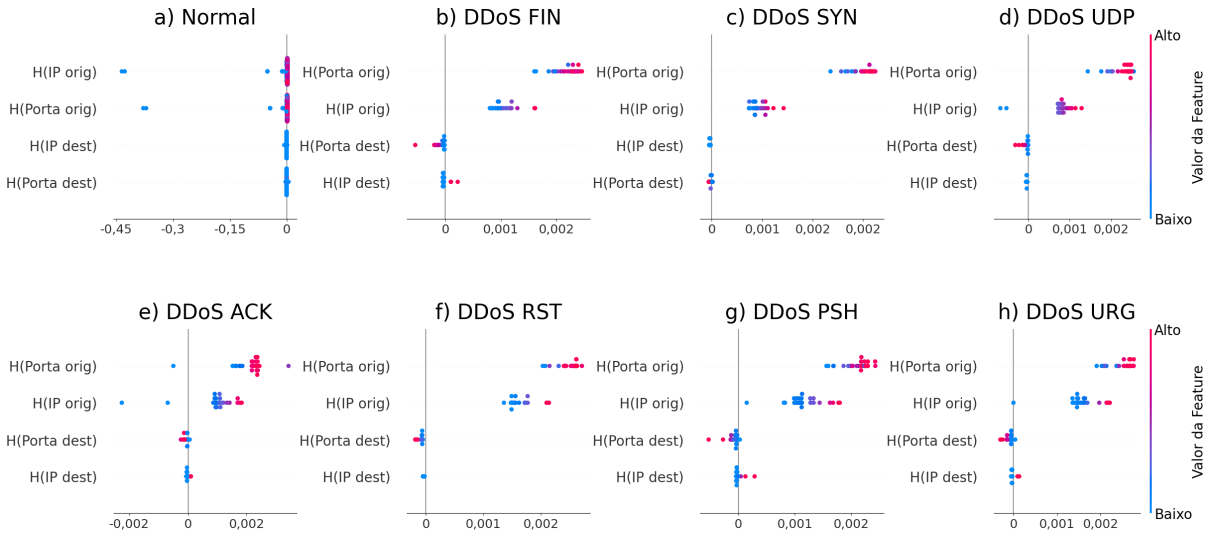

Figura 35 – Gráfico de enxame do quarto dia do conjunto de dados Orion2026 separado por categoria antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Em todos os cenários de ataque, as entropias de porta e IP de destino concentram-se em valores de SHAP próximos ao valor esperado, enquanto as variáveis remanescentes apresentam uma tendência positiva. Ademais, a escala do eixo horizontal nos três gráficos revela uma baixa amplitude de variação na saída do modelo para o tráfego espúrio, situando-se entre 10^{-3} e 10^{-2}. Esse estreitamento da faixa de valores de SHAP sugere uma proximidade da modelagem de ataques ao limiar de decisão do classificador, o que justificaria o incremento na taxa de falsos positivos.

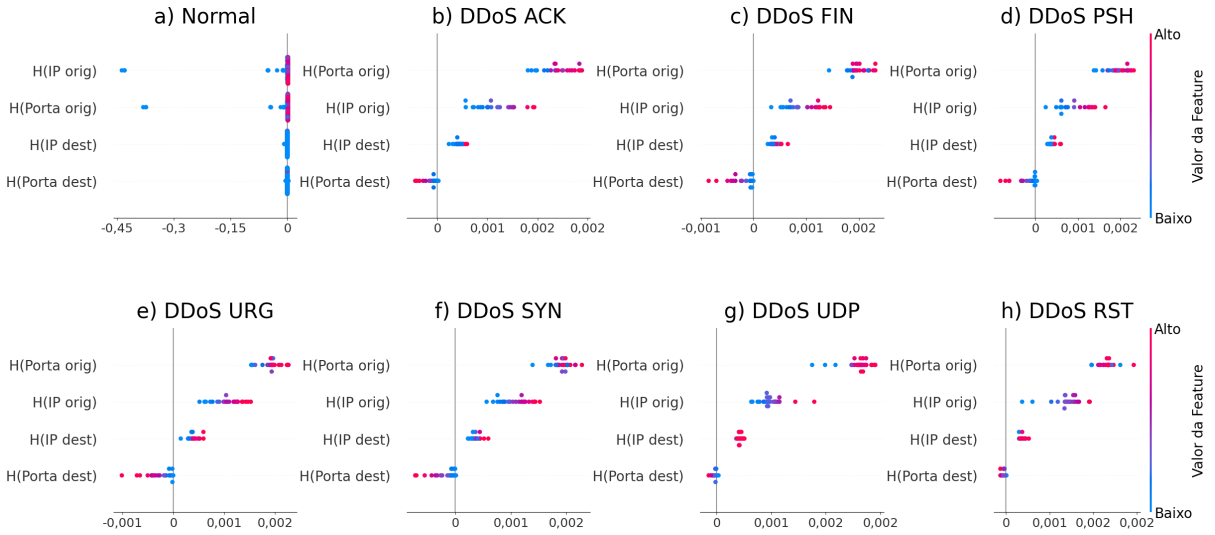

Figura 36 – Gráfico de enxame do quinto dia do conjunto de dados Orion2026 separado por categoria antes da otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Após o retreinamento do sistema mediante o descarte das entropias de destino, obteve-se uma modelagem mais estável e veloz, que equilibra o *trade-off* entre alarmes falsos e ataques indetectados. As novas métricas de desempenho são comparadas com as do sistema original na Tabela 6, evidenciando um incremento global nos índices de F1-Score, MCC e precisão. Destaca-se o avanço na precisão, que saltou da faixa de 90% no modelo original para valores entre 94,3% e 95,2% na versão otimizada, reflexo da redução na taxa de falsos positivos. Como consequência, o critério mais rigoroso de decisão da modelagem ocasiona a redução na métrica de revocação, que oscilava próxima a 99% e passou a situar-se na faixa de 97%. Sob a perspectiva prática da gerência de redes, essa dinâmica indica que o sistema otimizado passou a priorizar a mitigação de falsos alarmes para evitar a punição de usuários legítimos, ainda que ao custo de uma redução de aproximadamente 2 p.p. na revocação. Contudo, considerando o caráter inundativo dos ataques de negação de serviço, essa margem residual tende a não comprometer a eficiência do NIDS, visto que a maior parcela do volume malicioso ainda é detectada, neutralizando sua capacidade de congestionar os serviços da rede.

Tabela 6 – Comparação de desempenho do NIDS baseado em β-VAE original e otimizado para os três dias de teste do conjunto de dados Orion2026. **Fonte:** elaboração do próprio autor.

Métrica	Original			Otimizado		
	Dia 3	Dia 4	Dia 5	Dia 3	Dia 4	Dia 5
F1-Score	0,947	0,947	0,946	0,956	0,964	0,974
MCC	0,928	0,928	0,928	0,940	0,950	0,966
Precisão	0,906	0,903	0,898	0,943	0,951	0,952
Revocação	0,991	0,996	0,999	0,971	0,977	0,978
Tempo de inferência (segundos)	0,910	0,670	0,790	0,760	0,650	0,670

O comportamento da nova modelagem é ilustrado nas Figuras 37, 38 e 39. Para o tráfego espúrio dos três dias, detalhado nos subitens de “b” a “h”, observa-se uma inclinação das características remanescentes em se concentrarem em valores elevados de SHAP. No terceiro dia, sob a cardinalidade 1-1, a entropia da porta de origem apresenta um menor desvio em relação ao valor esperado. Essa dinâmica está associada ao menor volume de tráfego gerado no fluxo entre um único *host* de origem e um de destino, o que descarta a necessidade de uma nova rodada de otimização para a remoção dessa variável.

Por sua vez, o tráfego legítimo (subitem “a”) de todos os dias evidencia a concentração dos dados sobre o valor esperado. Essa combinação de fatores permite que o sistema opere com um limiar de decisão otimizado, reduzindo a classificação incorreta de ocorrências legítimas. Consequentemente, mitiga-se a indisponibilidade de serviços para usuários autorizados, o que minimiza a sobrecarga operacional decorrente de chamados de suporte e solicitações de restabelecimento de acesso.

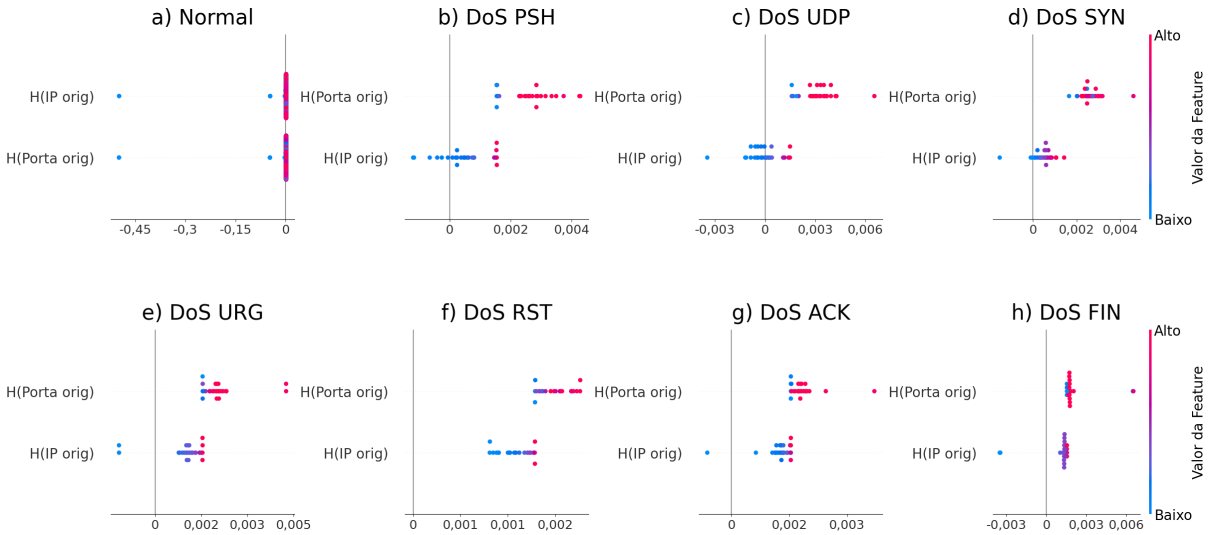

Figura 37 – Gráfico de enxame do terceiro dia do conjunto de dados Orion2026 separado por categoria após a otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

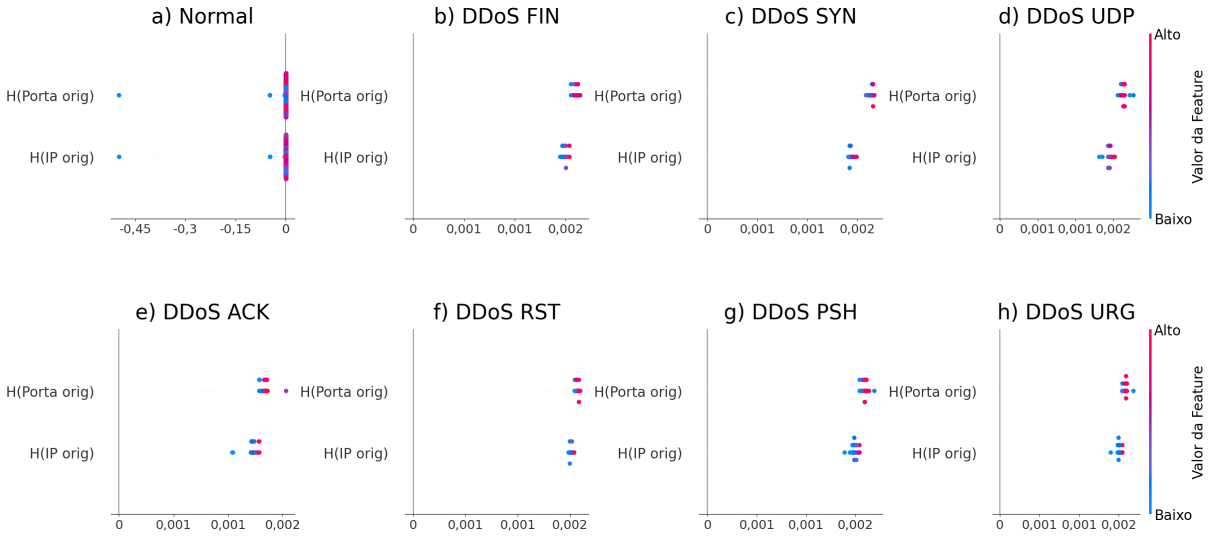

Figura 38 – Gráfico de enxame do quarto dia do conjunto de dados Orion2026 separado por categoria após a otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

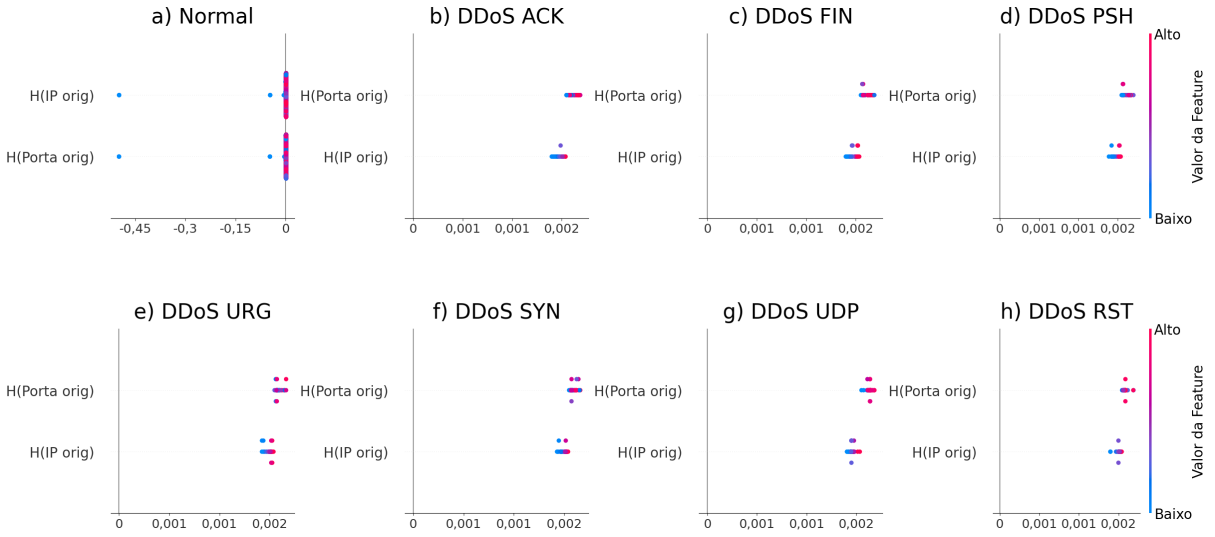

Figura 39 – Gráfico de enxame do quinto dia do conjunto de dados Orion2026 separado por categoria após a otimização do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Temporalmente, o maior ganho de estabilidade decorrente da otimização é observado para o ataque DoS UDP, executado no terceiro dia. No modelo original, ilustrado na Figura 40a, as entropias de porta e IP de destino mantêm-se alinhadas ao valor esperado ao longo de todo o período. As entropias de porta e IP de origem iniciam a janela temporal em patamares antagônicos, assumindo valores de SHAP positivos e negativos, respectivamente, e convergindo para o valor esperado ao final do dia. Como consequência,

a saída do modelo mantém-se próxima ao valor esperado durante todo o dia, apesar de crescer progressivamente.

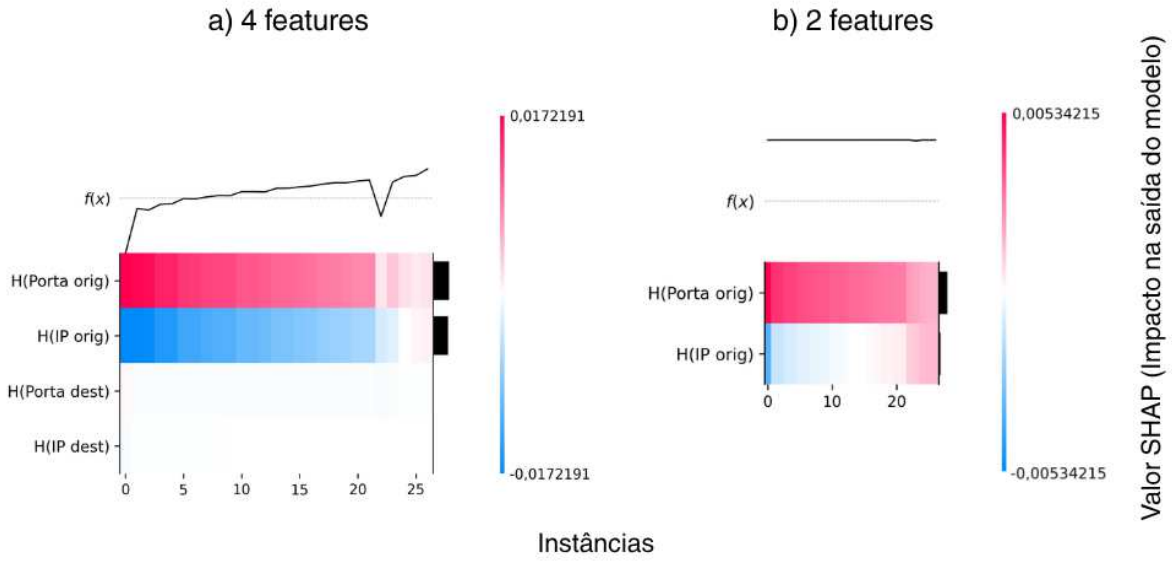

Figura 40 – Impactos temporais SHAP das características para o ataque UDP do conjunto de dados Orion2026 nas configurações exploradas do NIDS baseado em β-VAE. **Fonte:** elaboração do próprio autor.

Para o sistema otimizado, a entropia do IP de origem concentra-se majoritariamente sobre o valor esperado de SHAP. Há uma compensação da entropia da porta de origem, que passa a se situar predominantemente em valores positivos de SHAP, criando um balanceamento de influências das características. Esse novo equilíbrio estabiliza a saída do modelo acima do valor esperado de forma consistente, reduzindo a ocorrência de falsos negativos para esse ataque.

5.8 Discussão

A engenharia de características guiada por explicações de XAI exige rigor metodológico para evitar o sobreajuste e o subajuste do modelo caixa-preta. Se o conjunto de dados utilizado para a geração de explicações não refletir os cenários operacionais reais que o modelo enfrentará em ambiente de produção, a metodologia de otimização tenderá ao sobreajuste, comprometendo a capacidade de generalização do classificador diante de tráfego inédito. Em contrapartida, a busca inflexível pela perfeita separação e disposição das classes nas explicações pode resultar no descarte de atributos informativos, induzindo ao subajuste.

Para mitigar esses extremos, a abordagem proposta depende de um balanceamento pragmático. A remoção de atributos não deve almejar a erradicação de toda contradição no espaço de características, mas concentrar-se na poda daquelas variáveis que provocam distorções persistentes nas predições do modelo. Ademais, tendo em vista o custo computacional exponencial do SHAP, estratégias de subamostragem tornam-se essenciais no fluxo de trabalho. Nesse contexto, a técnica de subamostragem baseada em agrupamento proposta neste trabalho desempenha um papel crucial na metodologia de otimização, pois concilia a alocação reduzida de recursos computacionais com a preservação da variabilidade e da representatividade estatística do tráfego.

O experimento conduzido no caso de estudo base, que compreende o cenário CIC-DDoS2019 e o modelo 1D-CNN-GAN, revelou que as entropias de IP de origem e de destino tendiam a induzir falsos positivos. Em contrapartida, a entropia da porta de destino apresentou maior impacto para discriminar corretamente o tráfego legítimo e o espúrio. A retenção dessa característica resultou no aumento do MCC de $0,804$ para $0,851$ $(+0,047)$, além de reduzir o número de falsos negativos em 26,6%. A extensão dos experimentos para um cenário alternativo, utilizando o conjunto de dados CSE-CIC-IDS2018, atesta a capacidade de generalização da metodologia de forma agnóstica ao *dataset*. Considerando as quatro entropias originais nestes dados, os falsos negativos estavam associados às flutuações temporais das características em torno do valor esperado de SHAP, o que mascarava a assinatura do ataque. A redução do espaço de características para a entropia da porta de destino atenuou essas oscilações na fronteira de decisão, elevando o MCC de $0,703$ para $0,916$ $(+0,213)$, o que elimina 71,7% da distância que separava o modelo original do desempenho ideal, e reduzindo o número de falsos negativos em 92,3%. Esse resultado valida que a seleção do atributo representativo para a assinatura volumétrica reduz ruídos estocásticos do modelo, gerando modelagens mais estáveis para o sistema baseado em série temporal e redes GAN.

As explicações do sistema baseado em 1D-CNN-GAN otimizado para os cenários do CIC revelaram um comportamento mais consistente da característica mantida ao longo de diferentes registros de tempo da janela de série temporal para ambos os conjuntos de

dados. Para o conjunto de dados CIC-DDoS2019, os dois segundos mais influentes da janela deslizante sempre exibiram um impacto correto na saída do modelo, enquanto os segundos 8 e 10 consistentemente prejudicaram as predições. O modelo treinado com o conjunto de dados CSE-CIC-IDS2018 apresentou inconsistências para os passos de tempo 10, 11 e 13 da característica mantida. Nesses casos, uma técnica mais refinada para determinar o tamanho da janela deslizante da série temporal poderia ser aplicada para alcançar resultados ainda melhores. Além disso, o emprego de apenas uma característica não demonstrou sinais de decaimento de influência.

Considerando a arquitetura do β-VAE, cuja análise de tráfego decorre segundo a segundo, reforça-se a independência arquitetural da otimização proposta. Nos cenários CIC-DDoS2019 e CSE-CIC-IDS2018, o descarte das características com comportamento indesejado nas explicações de SHAP e a retenção exclusiva da entropia de porta de destino elevaram o MCC de $0,729$ e $0,786$ para $0,845$ e $0,940$, respectivamente. As análises temporais via SHAP indicam que os falsos negativos ainda presentes na modelagem otimizada decorrem da concentração de amostras espúrias em valores negativos de SHAP. Esse comportamento reflete períodos específicos de execução em que os ataques apresentam baixo volume de tráfego, comumente no início ou finalização do ataque distribuído, aproximando temporariamente o perfil estatístico das anomalias à assinatura do tráfego legítimo.

Os ensaios no conjunto Orion2026 demonstram a estabilidade da metodologia proposta diante de variações na cardinalidade entre atacantes e vítimas. As explicações do modelo para os cenários unificado (1-1), distribuído quanto a atacantes (N-1) e duplamente distribuído (N-N) revelaram que as entropias de IP e porta de destino atuavam originalmente como indutoras de falsos negativos. O descarte dessas características não apenas elevou a métrica MCC em média 0,024, de 0,928 para 0,952, mas também otimizou o balanceamento entre falsas classificações, reduzindo a taxa de falsos positivos ao custo de um incremento marginal de ataques indetectados. Considerando a natureza inundatória das anomalias avaliadas, esse *trade-off* configura um sistema de gerência com menor carga operacional, diminuindo a abertura de chamados de suporte decorrentes do bloqueio de tráfego benigno.

A exclusão de variáveis com comportamento indesejado apontado por XAI atenua a sobrecarga computacional de processamento dos dados de entrada do NIDS, além de aumentar o desempenho discriminativo. Essa economia de recursos é crucial em cenários de saturação de conexão, como os ataques de negação de serviço considerados, pois garante a disponibilidade do *hardware* para a execução de contramedidas automáticas e serviços de infraestrutura e provisão de *software*.

6 CONCLUSÃO

Neste trabalho foi apresentada uma metodologia que explora o potencial das explicações aditivas de Shapley para melhorar o desempenho de sistemas de detecção de intrusão. Esse método promove um sistema final explicável, bem como um processo de otimização transparente baseado em SHAP. Primeiro, o conjunto de dados é analisado sob o escopo da Análise Exploratória de Dados, buscando por correlações entre as características de tráfego. Uma vez que a Análise Exploratória de Dados é limitada aos dados e não é suficiente para refletir os padrões modelados pelo sistema, SHAP foi empregado para explicar o processo decisório do modelo.

Para lidar com o uso intenso de recursos desse algoritmo, um método de amostragem não supervisionado foi proposto a fim de manter a representatividade dos dados. As versões subamostradas dos conjuntos de dados CIC-DDoS2019, CSE-CIC-IDS2018 e Orion2026 alimentaram a implementação do KernelSHAP, o que revelou explicações estáticas e temporais dos modelos caixas-pretas separadas por tipo de tráfego. A seleção transparente de características foi conduzida por meio da análise de seus impactos na modelagem do tráfego normal e na detecção de ataques, bem como do decaimento de influência e do efeito de atraso temporal. Após o retreino usando as características apontadas como desejáveis, o sistema de detecção de intrusão foi explicado uma última vez para aumentar sua confiabilidade.

O estudo de caso base, implementado utilizando 1D-CNN-GAN otimizado para o conjunto de dados CIC-DDoS2019, apresentou um avanço absoluto de 0,047 no *Matthews Correlation Coefficient* e de 3,6 p.p. no F1-Score (ganhos relativos de 5,8% e 4,2%), em relação à versão original do sistema. O número de falsos negativos caiu de 693 para 509, uma queda de 26,6%, o que contribuiu para um avanço de 6,0 p.p. na métrica de revocação. O modelo tornou-se 0,67 segundos mais lento em predizer o dia de teste, o que pode ser atribuído ao aumento da arquitetura da rede discriminadora ou da dimensão da janela deslizante. De toda forma, esse atraso adiciona aproximadamente 88 μs na predição de cada janela temporal atualizada a cada segundo, o que pode ser considerado negligenciável da perspectiva dos prejuízos causados por esse período adicional de ataques de inundação. Para o conjunto CSE-CIC-IDS2018, o modelo otimizado obteve um avanço absoluto de 0,213 na métrica MCC e de 19,9 p.p. no F1-Score, o que corresponde a ganhos relativos de 30,3% e 27,4%. O número de falsos negativos diminuiu 92,3%, de 1.159 para 89 ocorrências, o que se refletiu em um avanço de 32,5 p.p. na métrica de revocação.

A validação da metodologia proposta em uma arquitetura de rede neural distinta revelou comportamento similar para os conjuntos de dados CIC-DDoS2019 e CSE-CIC-IDS2018, culminando na manutenção da entropia de porta de destino. Para o primeiro

dataset, obtiveram-se ganhos de 0,116 em MCC (valor absoluto) e de 12,5 p.p. em F1-Score, equivalentes a acréscimos relativos de 15,9% e 17,3%. Ocorrências legítimas incorretamente classificadas como espúrias diminuíram de 1.966 para 165, uma queda de 91,6%, contribuindo para um avanço de 34,4 p.p. na métrica da precisão, que saltou de 0,591 para 0,935 (um ganho relativo de 58,2%). Concomitantemente, o sistema atingiu melhoras de 0,154 em MCC e de 11,1 p.p. em F1-Score para o segundo conjunto de dados (ganhos relativos de 19,6% e 13,0%), reflexo da diminuição da taxa de falsos negativos de 24,6% para 3,7%, recuo de 20,9 p.p.

Estendeu-se, ainda, a análise da aplicabilidade da otimização proposta para cenários de diferentes cardinalidades entre atacantes e vítimas. Utilizando a base de dados Orion2026, o NIDS baseado em β-VAE apresentou uma modelagem consistente para as variações de ataques, capturando o comportamento espúrio ao amparar-se nas entropias de IP e porta de origem. O descarte das características de entropia de destino possibilitou o treinamento de um sistema que, em média, atingiu MCC 0,024 maior e F1-Score 1,8 p.p. maior, ganhos relativos de 2,6% e 1,9%. Além disso, a versão otimizada contou com menor punição de usuários legítimos.

A otimização baseada em SHAP impacta diretamente a gerência de redes ao mitigar falsos alarmes e reduzir a saturação operacional associada à triagem de incidentes de segurança. A identificação e o descarte de características redundantes, irrelevantes ou enviesadas refinam a modelagem do tráfego e os limites de decisão dos sistemas de detecção de intrusão, resultando na queda das taxas de falsos positivos e negativos. Essa maior assertividade reduz a punição injusta de usuários legítimos e minimiza o volume de chamados técnicos e correções manuais na liberação de tráfego bloqueado erroneamente. Dessa forma, os administradores de rede conseguem concentrar recursos humanos e analíticos em ameaças reais, otimizando o fluxo de trabalho e a governança de TI. Sob a perspectiva do consumo de recursos computacionais, a metodologia introduz um compromisso de balanceamento entre *overhead* de processamento, robustez de segurança e confiabilidade em sistemas críticos automáticos.

Apesar dos ganhos obtidos, a metodologia proposta apresenta limitações que devem ser destacadas. A otimização baseada em SHAP demanda a atuação de um gerente de processo capacitado para interpretar os gráficos de enxame e de mapa de calor, o que confere ao processo um caráter essencialmente visual. Essa dependência da interpretação visual acarreta o risco de sobreajuste ou subajuste no modelo retreinado. Ademais, o descarte excessivo de características pode comprometer a capacidade do sistema de detectar ataques do tipo *zero-day*, visto que informações relevantes para a generalização do modelo podem ser eliminadas precocemente. Por fim, o cálculo das explicações de Shapley é computacionalmente custoso, de modo que sua aplicação durante o treinamento do sistema, anterior à implantação em produção, agrega sobrecarga ao processo. Em um cenário de

sistema online ou de aprendizado incremental, o uso do SHAP poderia introduzir um atraso adicional àquele já inerente ao cálculo da entropia de fluxos IP a cada segundo.

Como trabalhos futuros, pretende-se automatizar o processo de otimização por meio da aplicação do método do cotovelo sobre as importâncias individuais das características. Essa estratégia visa identificar o ponto de corte a partir do qual a influência das *features* deixa de ser expressiva para a modelagem. Deve-se, ainda, investigar a incorporação da otimização baseada em SHAP em NIDS que empregam aprendizado online ou incremental, capazes de lidar com ataques *zero-day* e desvio de conceito. Tais sistemas tendem a demandar a reavaliação periódica das características selecionadas à medida que o tráfego evolui. Isso impulsiona a investigação de estratégias que viabilizem essa reavaliação sem comprometer o desempenho computacional nem a latência de detecção em tempo real.

REFERÊNCIAS

[1] SOOD, K. et al. A tutorial on next generation heterogeneous IoT networks and node authentication. *IEEE Internet of Things Magazine*, v. 4, n. 4, p. 120–126, 2021. Disponível em: <https://doi.org/10.1109/IOTM.001.2100115>.

[2] ZHANG, Y. et al. Improving SD-WAN resilience: From vertical handoff to WAN-Aware MPTCP. *IEEE Transactions on Network and Service Management*, v. 18, n. 1, p. 347–361, 2021. Disponível em: <https://doi.org/10.1109/TNSM.2021.3052471>.

[3] BORGIANNI, L.; ADAMI, D.; GIORDANO, S. Towards 6G: Adaptive and resilient network management for hybrid terrestrial-satellite systems. In: *NOMS 2025-2025 IEEE Network Operations and Management Symposium*. [S.l.]: [s.n.], 2025. p. 1–4. Disponível em: <https://doi.org/10.1109/NOMS57970.2025.11073601>.

[4] RAJESH, K.; VETRIVELAN, P. Comprehensive analysis on 5G and 6G wireless network security and privacy. *Telecommunication Systems*, Springer, New York, v. 88, n. 2, p. 52, 2025. Disponível em: <https://doi.org/10.1007/s11235-025-01282-2>.

[5] EKE, C. I.; NORMAN, A. A.; MULENGA, M. Machine learning approach for detecting and combating bring your own device (BYOD) security threats and attacks: a systematic mapping review. *Artificial Intelligence Review*, Springer, New York, v. 56, n. 8, p. 8815–8858, 2023. Disponível em: <https://doi.org/10.1007/s10462-022-10382-3>.

[6] MEKRACHE, A.; KSENTINI, A.; VERIKOUKIS, C. Machine learning in FCAPS: Toward enhanced beyond 5G network management. *IEEE Communications Surveys & Tutorials*, v. 26, n. 4, p. 2769–2797, 2024. Disponível em: <https://doi.org/10.1109/COMST.2024.3395414>.

[7] BARNAWI, A. et al. A systematic analysis of deep learning methods and potential attacks in internet-of-things surfaces. *Neural Computing and Applications*, Springer, New York, v. 35, n. 25, p. 18293–18308, 2023. Disponível em: <https://doi.org/10.1007/s00521-023-08634-6>.

[8] LIANG, X.; XU, Y. A novel framework to identify cybersecurity challenges and opportunities for organizational digital transformation in the cloud. *Computers & Security*, v. 151, p. 104339, 2025. Disponível em: <https://doi.org/10.1016/j.cose.2025.104339>.

[9] LYU, M.; GHARAKHEILI, H. H.; SIVARAMAN, V. A survey on enterprise network security: Asset behavioral monitoring and distributed attack detection. *IEEE Access*, v. 12, p. 89363–89383, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2024.3419068>.

[10] RREDDY, M. V. K.; LATHIGARA, A.; REDDY, M. K. Attack detection in smart home IoT networks: A survey on challenges, methods and analysis. In: CHENG, X. (Ed.). *Broadband Communications, Networks, and Systems*. Cham: Springer Nature Switzerland, 2025. p. 310–319. ISBN 978-3-031-81168-5. Disponível em: <https://doi.org/10.1007/978-3-031-81168-5_29>.

[11] CHERFI, S.; LEMOUARI, A.; BOULAICHE, A. MLP-based intrusion detection for securing IoT networks. *Journal of Network and Systems Management*, Springer, New York, v. 33, n. 1, p. 20, 2025. Disponível em: <https://doi.org/10.1007/s10922-024-09889-7>.

[12] PROENÇA, M. L. et al. Baseline to help with network management. In: ASCENSO, J. et al. (Ed.). *e-Business and Telecommunication Networks*. Dordrecht: Springer Netherlands, 2006. p. 158–166. ISBN 978-1-4020-4761-9. Disponível em: <https://doi.org/10.1007/1-4020-4761-4_12>.

[13] RJOUB, G. et al. A survey on explainable artificial intelligence for cybersecurity. *IEEE Transactions on Network and Service Management*, v. 20, n. 4, p. 5115–5140, 2023. Disponível em: <https://doi.org/10.1109/TNSM.2023.3282740>.

[14] KOKILA, M. K.; KONDA, S. R. DeepSDN: Deep learning based software defined network model for cyberthreat detection in IoT network. *ACM Transactions on Internet Technology*, Association for Computing Machinery, New York, NY, USA, v. 26, n. 1, p. 1–29, 2025. Disponível em: <https://doi.org/10.1145/3737875>.

[15] MOENS, P. et al. Toward context-aware anomaly detection for AIOps in microservices using dynamic knowledge graphs. *IEEE Transactions on Network and Service Management*, v. 23, p. 1970–1988, 2026. Disponível em: <https://doi.org/10.1109/TNSM.2026.3652304>.

[16] TSIMENIDIS, S.; LAGKAS, T.; RANTOS, K. Deep learning in IoT intrusion detection. *Journal of Network and Systems Management*, Springer, New York, v. 30, n. 1, p. 8, 2022. Disponível em: <https://doi.org/10.1007/s10922-021-09621-9>.

[17] TELLACHE, A. et al. Multi-agent reinforcement learning-based network intrusion detection system. In: *NOMS 2024-2024 IEEE Network Operations and Management Symposium*. [S.l.]: [s.n.], 2024. p. 1–9. Disponível em: <https://doi.org/10.1109/NOMS59830.2024.10575541>.

[18] RUFFO, V. G. da S. et al. Anomaly and intrusion detection using deep learning for software-defined networks: A survey. *Expert Systems with Applications*, v. 256, p. 124982, 2024. Disponível em: <https://doi.org/10.1016/j.eswa.2024.124982>.

[19] MARY, D. S. et al. Network intrusion detection: An optimized deep learning approach using big data analytics. *Expert Systems with Applications*, Elsevier, v. 251, p. 123919, 2024. Disponível em: <https://doi.org/10.1016/j.eswa.2024.123919>.

[20] SCARANTI, G. F. et al. Unsupervised online anomaly detection in software defined network environments. *Expert Systems with Applications*, v. 191, p. 116225, 2022. Disponível em: <https://doi.org/10.1016/j.eswa.2021.116225>.

[21] LENT, D. M. B. et al. A gated recurrent unit deep learning model to detect and mitigate distributed denial of service and portscan attacks. *IEEE Access*, IEEE, v. 10, p. 73229–73242, 2022. Disponível em: <https://doi.org/10.1109/ACCESS.2022.3190008>.

[22] FERNANDES, G. et al. A comprehensive survey on network anomaly detection. *Telecommunication Systems*, Springer, New York, v. 70, n. 3, p. 447–489, 2019. Disponível em: <https://doi.org/10.1007/s11235-018-0475-8>.

[23] ASSIS, M. V. O. de et al. Holt-Winters statistical forecasting and ACO metaheuristic for traffic characterization. In: *2013 IEEE International Conference on Communications (ICC)*. [s.n.], 2013. p. 2524–2528. Disponível em: <https://doi.org/10.1109/ICC.2013.6654913>.

[24] RUFFO, V. G. da S. et al. f-AnoGAN for unsupervised attack detection in SDN environment. *IEEE Transactions on Network Science and Engineering*, v. 12, n. 4, p. 3271–3285, 2025. Disponível em: <https://doi.org/10.1109/TNSE.2025.3558936>.

[25] ASSIS, M. V. O. de et al. A GRU deep learning system against attacks in software defined networks. *Journal of Network and Computer Applications*, v. 177, p. 102942, 2021. Disponível em: <https://doi.org/10.1016/j.jnca.2020.102942>.

[26] ZACARON, A. M. et al. Generative adversarial network models for anomaly detection in software-defined networks. *Journal of Network and Systems Management*, Springer, New York, v. 32, n. 4, p. 93, 2024. Disponível em: <https://doi.org/10.1007/s10922-024-09867-z>.

[27] ANDRESINI, G. et al. ROULETTE: A neural attention multi-output model for explainable network intrusion detection. *Expert Systems with Applications*, Elsevier, Amsterdam, v. 201, p. 117144, 2022. Disponível em: <https://doi.org/10.1016/j.eswa.2022.117144>.

[28] MUTALIB, N. H. A. et al. Explainable deep learning approach for advanced persistent threats (APTs) detection in cybersecurity: a review. *Artificial Intelligence Review*, Springer, New York, v. 57, n. 11, p. 297, 2024. Disponível em: <https://doi.org/10.1007/s10462-024-10890-4>.

[29] AL, S.; SAGIROGLU, S. Explainable artificial intelligence models in intrusion detection systems. *Engineering Applications of Artificial Intelligence*, v. 144, p. 110145, 2025. Disponível em: <https://doi.org/10.1016/j.engappai.2025.110145>.

[30] LUNDBERG, S. M.; LEE, S.-I. A unified approach to interpreting model predictions. In: *Proceedings of the 31st International Conference on Neural Information Processing Systems*. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 4768–4777. ISBN 978-15-108-6096-4.

[31] IRÉNÉE, M. et al. XTS: A hybrid framework to detect DNS-Over-HTTPS tunnels based on XGBoost and cooperative game theory. *Mathematics*, v. 11, n. 10, p. 2372, 2023. Disponível em: <https://doi.org/10.3390/math11102372>.

[32] MAHMOUD, M. M.; YOUSSEF, Y. O.; ABDEL-HAMID, A. A. XI2S-IDS: An explainable intelligent 2-stage intrusion detection system. *Future Internet*, v. 17, n. 1, p. 25, 2025. Disponível em: <https://doi.org/10.3390/fi17010025>.

[33] KESHK, M. et al. An explainable deep learning-enabled intrusion detection framework in IoT networks. *Information Sciences*, v. 639, p. 119000, 2023. Disponível em: <https://doi.org/10.1016/j.ins.2023.119000>.

[34] AL-ESSA, M. et al. PANACEA: a neural model ensemble for cyber-threat detection. *Machine Learning*, Springer, New York, v. 113, n. 8, p. 5379–5422, 2024. Disponível em: <https://doi.org/10.1007/s10994-023-06470-2>.

[35] ARRECHE, O.; BIBERS, I.; ABDALLAH, M. A two-level ensemble learning framework for enhancing network intrusion detection systems. *IEEE Access*, v. 12, p. 83830–83857, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2024.3407029>.

[36] CHINU; BANSAL, U. Fair XIDS: Ensuring fairness and transparency in intrusion detection models. *Concurrency and Computation: Practice and Experience*, v. 36, n. 25, p. e8268, 2024. Disponível em: <https://doi.org/10.1002/cpe.8268>.

[37] SHTAYAT, M. M. et al. An explainable ensemble deep learning approach for intrusion detection in industrial internet of things. *IEEE Access*, v. 11, p. 115047–115061, 2023. Disponível em: <https://doi.org/10.1109/ACCESS.2023.3323573>.

[38] SHARAFALDIN, I. et al. Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. In: *2019 International Carnahan Conference on Security Technology (ICCST)*. [S.l.]: [s.n.], 2019. p. 1–8. Disponível em: <https://doi.org/10.1109/CCST.2019.8888419>.

[39] SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. 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)*. Funchal: SciTePress, 2018. p. 108–116. Disponível em: <https://doi.org/10.5220/0006639801080116>.

[40] ORION 2026: An IP Flow Dataset for Volumetric Traffic Analysis and DDoS Intrusion Detection. 2026. Acesso em: 6 jun. 2026. Disponível em: <https://www.uel.br/grupos/orion/datasets/orion-2026.html>.

[41] ABDULGANIYU, O. H.; TCHAKOUCHT, T. A.; SAHEED, Y. K. A systematic literature review for network intrusion detection system (IDS). *International Journal of Information Security*, Springer, New York, v. 22, n. 5, p. 1125–1162, 2023. Disponível em: <https://doi.org/10.1007/s10207-023-00682-2>.

[42] SOUZA, C. A. de et al. Intrusion detection and prevention in fog based IoT environments: A systematic literature review. *Computer Networks*, v. 214, p. 109154, 2022. Disponível em: <https://doi.org/10.1016/j.comnet.2022.109154>.

[43] YANG, Z. et al. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, v. 116, p. 102675, 2022. Disponível em: <https://doi.org/10.1016/j.cose.2022.102675>.

[44] PROENÇA, M. L. et al. The hurst parameter for digital signature of network segment. In: SOUZA, J. N. de; DINI, P.; LORENZ, P. (Ed.). *Telecommunications and Networking - ICT 2004*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2004. p. 772–781. ISBN 978-3-540-27824-5. Disponível em: <https://doi.org/10.1007/978-3-540-27824-5_103>.

[45] PROENÇA, M. L.; ZARPELÃO, B. B.; MENDES, L. S. Anomaly detection for network servers using digital signature of network segment. In: *Advanced Industrial Conference on Telecommunications/Service Assurance with Partial and Intermittent Resources Conference/E-Learning on Telecommunications Workshop (AICT/SAPIR/ELETE'05)*. [S.l.]: [s.n.], 2005. p. 290–295. Disponível em: <https://doi.org/10.1109/AICT.2005.26>.

[46] NOVAES, M. P. et al. Adversarial deep learning approach detection and defense against DDoS attacks in SDN environments. *Future Generation Computer Systems*, v. 125, p. 156–167, 2021. Disponível em: <https://doi.org/10.1016/j.future.2021.06.047>.

[47] CERVANTES, J. et al. A comprehensive survey on support vector machine classification: Applications, challenges and trends. *Neurocomputing*, v. 408, p. 189–215, 2020. Disponível em: <https://doi.org/10.1016/j.neucom.2019.10.118>.

[48] MIENYE, I. D.; JERE, N. A survey of decision trees: Concepts, algorithms, and applications. *IEEE Access*, v. 12, p. 86716–86727, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2024.3416838>.

[49] MIENYE, I. D.; SUN, Y. A survey of ensemble learning: Concepts, algorithms, applications, and prospects. *IEEE Access*, v. 10, p. 99129–99149, 2022. Disponível em: <https://doi.org/10.1109/ACCESS.2022.3207287>.

[50] OZKAN-OKAY, M. et al. A comprehensive survey: Evaluating the efficiency of artificial intelligence and machine learning techniques on cyber security solutions. *IEEE Access*, v. 12, p. 12229–12256, 2024. Disponível em: <https://doi.org/10.1109/ACCESS.2024.3355547>.

[51] GOODFELLOW, I. J. et al. Generative adversarial nets. *Advances in Neural Information Processing Systems*, v. 27, 2014. Disponível em: <https://doi.org/10.48550/arXiv.1406.2661>.

[52] BISHOP, C. M.; BISHOP, H. *Deep learning: Foundations and concepts*. Cham: Springer Nature, 2023. ISBN 978-3031454677.

[53] GOODFELLOW, I. et al. *Deep learning*. Cambridge, MA: MIT Press, 2016. v. 1. ISBN 978-0262035613.

[54] MURPHY, K. P. *Probabilistic machine learning: an introduction*. Cambridge, MA: MIT Press, 2022. ISBN 978-0262046824.

[55] CAI, Z. et al. Generative adversarial networks: A survey toward private and secure applications. *ACM Computing Surveys (CSUR)*, Association for Computing Machinery, New York, NY, USA, v. 54, n. 6, p. 1–38, 2021. Disponível em: <https://doi.org/10.1145/3459992>.

[56] LIN, H. et al. How generative adversarial networks promote the development of intelligent transportation systems: A survey. *IEEE/CAA Journal of Automatica Sinica*, IEEE, v. 10, n. 9, p. 1781–1796, 2023. Disponível em: <https://doi.org/10.1109/JAS.2023.123744>.

[57] GUI, J. et al. A review on generative adversarial networks: Algorithms, theory, and applications. *IEEE Transactions on Knowledge and Data Engineering*, v. 35, n. 4, p. 3313–3332, 2023. Disponível em: <https://doi.org/10.1109/TKDE.2021.3130191>.

[58] SOUZA, V. L. T. de et al. A review on generative adversarial networks for image generation. *Computers & Graphics*, Elsevier, Amsterdam, v. 114, p. 13–25, 2023. Disponível em: <https://doi.org/10.1016/j.cag.2023.05.010>.

[59] RADFORD, A.; METZ, L.; CHINTALA, S. Unsupervised representation learning with deep convolutional generative adversarial networks. *arXiv preprint arXiv:1511.06434*, 2015. Disponível em: <https://doi.org/10.48550/arXiv.1511.06434>.

[60] MIRZA, M.; OSINDERO, S. Conditional generative adversarial nets. *arXiv preprint arXiv:1411.1784*, 2014. Disponível em: <https://doi.org/10.48550/arXiv.1411.1784>.

[61] ARJOVSKY, M.; CHINTALA, S.; BOTTOU, L. Wasserstein generative adversarial networks. In: PRECUP, D.; TEH, Y. W. (Ed.). *Proceedings of the 34th International Conference on Machine Learning*. [S.l.]: PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 214–223.

[62] LECUN, Y. et al. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, v. 86, n. 11, p. 2278–2324, 1998. Disponível em: <https://doi.org/10.1109/5.726791>.

[63] LIAO, H.; ZHOU, S. K.; LUO, J. Chapter 2 - deep learning basics. In: LIAO, H.; ZHOU, S. K.; LUO, J. (Ed.). *Deep Network Design for Medical Image Computing*. [S.l.]: Academic Press, 2023, (The MICCAI Society book Series). p. 11–23. ISBN 978-0-12-824383-1. Disponível em: <https://doi.org/10.1016/B978-0-12-824383-1.00009-5>.

[64] SZELISKI, R. *Computer vision: algorithms and applications*. Cham: Springer Nature, 2022. ISBN 978-3-030-34372-9. Disponível em: <https://doi.org/10.1007/978-3-030-34372-9>.

[65] DHILLON, A.; VERMA, G. K. Convolutional neural network: a review of models, methodologies and applications to object detection. *Progress in Artificial Intelligence*, Springer, New York, v. 9, n. 2, p. 85–112, 2020. Disponível em: <https://doi.org/10.1007/s13748-019-00203-0>.

[66] ZEILER, M. D.; FERGUS, R. Visualizing and understanding convolutional networks. In: FLEET, D. et al. (Ed.). *Computer Vision – ECCV 2014*. Cham: Springer International Publishing, 2014. p. 818–833. ISBN 978-3-319-10590-1. Disponível em: <https://doi.org/10.1007/978-3-319-10590-1_53>.

[67] ASSIS, M. V. O. de et al. Near real-time security system applied to SDN environments in IoT networks using convolutional neural network. *Computers & Electrical Engineering*, v. 86, p. 106738, 2020. Disponível em: <https://doi.org/10.1016/j.compeleceng.2020.106738>.

[68] ALZUBAIDI, L. et al. Review of deep learning: concepts, CNN architectures, challenges, applications, future directions. *Journal of Big Data*, Springer, New York, v. 8, n. 1, p. 53, 2021. Disponível em: <https://doi.org/10.1186/s40537-021-00444-8>.

[69] KINGMA, D. P.; WELLING, M. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013. Presented at the International Conference on Learning Representations (ICLR 2014). Disponível em: <https://doi.org/10.48550/arXiv.1312.6114>.

[70] HIGGINS, I. et al. β-VAE: Learning basic visual concepts with a constrained variational framework. In: *International Conference on Learning Representations (ICLR)*. [S.l.]: [s.n.], 2017.

[71] ADADI, A.; BERRADA, M. Peeking inside the black-box: A survey on explainable artificial intelligence (XAI). *IEEE Access*, v. 6, p. 52138–52160, 2018. Disponível em: <https://doi.org/10.1109/ACCESS.2018.2870052>.

[72] KÖK, I. et al. Explainable artificial intelligence (XAI) for internet of things: A survey. *IEEE Internet of Things Journal*, v. 10, n. 16, p. 14764–14779, 2023. Disponível em: <https://doi.org/10.1109/JIOT.2023.3287678>.

[73] SCHWALBE, G.; FINZEL, B. A comprehensive taxonomy for explainable artificial intelligence: a systematic survey of surveys on methods and concepts. *Data Mining and Knowledge Discovery*, Springer, New York, v. 38, n. 5, p. 3043–3101, 2024. Disponível em: <https://doi.org/10.1007/s10618-022-00867-8>.

[74] BARREDO ARRIETA, A. et al. Explainable artificial intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI. *Information Fusion*, v. 58, p. 82–115, 2020. Disponível em: <https://doi.org/10.1016/j.inffus.2019.12.012>.

[75] ALI, S. et al. Explainable artificial intelligence (XAI): What we know and what is left to attain trustworthy artificial intelligence. *Information Fusion*, v. 99, p. 101805, 2023. Disponível em: <https://doi.org/10.1016/j.inffus.2023.101805>.

[76] MONTAVON, G. et al. Explaining nonlinear classification decisions with deep Taylor decomposition. *Pattern Recognition*, v. 65, p. 211–222, 2017. Disponível em: <https://doi.org/10.1016/j.patcog.2016.11.008>.

[77] IVANOVS, M.; KADIKIS, R.; OZOLS, K. Perturbation-based methods for explaining deep neural networks: A survey. *Pattern Recognition Letters*, v. 150, p. 228–234, 2021. Disponível em: <https://doi.org/10.1016/j.patrec.2021.06.030>.

[78] KHAN, A. et al. Using permutation-based feature importance for improved machine learning model performance at reduced costs. *IEEE Access*, v. 13, p. 36421–36435, 2025. Disponível em: <https://doi.org/10.1109/ACCESS.2025.3544625>.

[79] BHAT, A.; ASSOA, A. S.; RAYCHOWDHURY, A. Gradient backpropagation based feature attribution to enable explainable-AI on the edge. In: *2022 IFIP/IEEE 30th International Conference on Very Large Scale Integration (VLSI-SoC)*. [S.l.]: IEEE, 2022. p. 1–6. Disponível em: <https://doi.org/10.1109/VLSI-SoC54400.2022.9939601>.

[80] SHRIKUMAR, A.; GREENSIDE, P.; KUNDAJE, A. Learning important features through propagating activation differences. In: *Proceedings of the 34th International Conference on Machine Learning (ICML)*. [S.l.]: PMLR, 2017. (Proceedings of Machine Learning Research, v. 70), p. 3145–3153.

[81] FRIEDMAN, J. H. Greedy function approximation: a gradient boosting machine. *The Annals of Statistics*, Institute of Mathematical Statistics, v. 29, n. 5, p. 1189–1232, 2001.

[82] GOLDSTEIN, A. et al. Peeking inside the black box: Visualizing statistical learning with plots of individual conditional expectation. *Journal of Computational and Graphical Statistics*, Taylor & Francis, v. 24, n. 1, p. 44–65, 2015. Disponível em: <https://doi.org/10.1080/10618600.2014.907095>.

[83] APLEY, D. W.; ZHU, J. Visualizing the effects of predictor variables in black box supervised learning models. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, Oxford University Press, v. 82, n. 4, p. 1059–1086, 2020. Disponível em: <https://doi.org/10.1111/rssb.12377>.

[84] GUIDOTTI, R. et al. Factual and counterfactual explanations for black box decision making. *IEEE Intelligent Systems*, v. 34, n. 6, p. 14–23, 2019. Disponível em: <https://doi.org/10.1109/MIS.2019.2957223>.

[85] KUPPA, A.; LE-KHAC, N.-A. Adversarial XAI methods in cybersecurity. *IEEE Transactions on Information Forensics and Security*, v. 16, p. 4924–4938, 2021. Disponível em: <https://doi.org/10.1109/TIFS.2021.3117075>.

[86] BARBADO, A.; CORCHO, Ó.; BENJAMINS, R. Rule extraction in unsupervised anomaly detection for model explainability: Application to OneClass SVM. *Expert Systems with Applications*, v. 189, p. 116100, 2022. Disponível em: <https://doi.org/10.1016/j.eswa.2021.116100>.

[87] ANDREWS, R.; DIEDERICH, J.; TICKLE, A. B. Survey and critique of techniques for extracting rules from trained artificial neural networks. *Knowledge-Based Systems*, Elsevier, v. 8, n. 6, p. 373–389, 1995. Disponível em: <https://doi.org/10.1016/0950-7051(96)81920-4>.

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

[89] LEE, J. H.; LANZA, S.; WERMTER, S. From neural activations to concepts: A survey on explaining concepts in neural networks. *Neurosymbolic Artificial Intelligence*, v. 1, p. NAI–240743, 2025. Disponível em: <https://doi.org/10.3233/NAI-240743>.

[90] JAIN, S.; WALLACE, B. C. Attention is not explanation. In: *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, 2019. p. 3543–3556. Disponível em: <https://doi.org/10.18653/v1/N19-1357>.

[91] WIEGREFFE, S.; PINTER, Y. Attention is not not explanation. In: *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*. Hong Kong, China: Association for Computational Linguistics, 2019. p. 11–20. Disponível em: <https://doi.org/10.18653/v1/D19-1002>.

[92] RIBEIRO, M. T.; SINGH, S.; GUESTRIN, C. "Why should I trust you?": Explaining the predictions of any classifier. In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*. New York, NY, USA: Association for Computing Machinery, 2016. (KDD '16), p. 1135–1144. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939778>.

[93] KOHAVI, R.; PROVOST, F. Glossary of terms. *Machine Learning*, v. 30, p. 271–274, 1998. Disponível em: <https://doi.org/10.1023/A:1017181826899>.

[94] BANERJEE, A. et al. Hypothesis testing, type I and type II errors. *Industrial Psychiatry Journal*, Medknow, v. 18, n. 2, p. 127–131, 2009. Disponível em: <https://doi.org/10.4103/0972-6748.62274>.

[95] CHRISTEN, P.; HAND, D. J.; KIRIELLE, N. A review of the F-measure: its history, properties, criticism, and alternatives. *ACM Computing Surveys*, Association for Computing Machinery, New York, NY, USA, v. 56, n. 3, p. 1–24, 2023. Disponível em: <https://doi.org/10.1145/3606367>.

[96] MATTHEWS, B. W. Comparison of the predicted and observed secondary structure of T4 phage lysozyme. *Biochimica et Biophysica Acta (BBA)-Protein Structure*, Elsevier, Amsterdam, v. 405, n. 2, p. 442–451, 1975. Disponível em: <https://doi.org/10.1016/0005-2795(75)90109-9>.

[97] CHICCO, D.; JURMAN, G. The advantages of the Matthews correlation coefficient (MCC) over F1 score and accuracy in binary classification evaluation. *BMC Genomics*, Springer, v. 21, n. 1, p. 6, 2020. Disponível em: <https://doi.org/10.1186/s12864-019-6413-7>.

[98] RUFFO, V. G. da S. et al. Generative adversarial networks to detect intrusion and anomaly in IP flow-based networks. *Future Generation Computer Systems*, v. 163, p. 107531, 2025. Disponível em: <https://doi.org/10.1016/j.future.2024.107531>.

[99] SHANNON, C. E. A mathematical theory of communication. *The Bell System Technical Journal*, v. 27, n. 3, p. 379–423, 1948. Disponível em: <https://doi.org/10.1002/j.1538-7305.1948.tb01338.x>.

[100] SCOTT, D. W. *Multivariate Density Estimation: Theory, Practice, and Visualization*. New York: John Wiley & Sons, 1992. ISBN 978-0471547709.

[101] FISHER, R. A.; TIPPETT, L. H. C. Limiting forms of the frequency distribution of the largest or smallest member of a sample. *Mathematical Proceedings of the Cambridge Philosophical Society*, v. 24, n. 2, p. 180–190, 1928. Disponível em: <https://doi.org/10.1017/S0305004100015681>.

[102] VASERSTEIN, L. N. Markov processes over denumerable products of spaces, describing large systems of automata. *Problemy Peredachi Informatsii*, Russian Academy of Sciences, v. 5, n. 3, p. 64–72, 1969.

TRABALHOS PUBLICADOS PELO AUTOR

Trabalhos publicados pelo autor durante o programa.

1. **Mateus Komarchesqui**, Daniel Matheus Brandão Lent, Vitor Gabriel da Silva Ruffo, Luiz Fernando Carvalho, Jaime Lloret, Mario Lemes Proença Jr., **Explainable AI Feature Selection in Generative Adversarial Networks System aiming to detect DDoS Attacks**, 2024 11th International Conference on Software Defined Systems (SDS), dezembro/2024, IEEE, 27-34, 10.1109/SDS64317.2024.10883903. (Qualis CC 2021-2024, A4).

2. Vitor Gabriel da Silva Ruffo, Daniel Matheus Brandão Lent, **Mateus Komarchesqui**, Vinícius Ferreira Schiavon, Marcos Vinicius Oliveira de Assis, Luiz Fernando Carvalho, Mario Lemes Proença Jr., **Anomaly and intrusion detection using deep learning for software-defined networks: A survey**, Expert Systems with Applications, dezembro/2024, Elsevier, 256, 124982, 10.1016/j.eswa.2024.124982. (Fator de Impacto 2025: 9,4; Qualis CC 2021-2024, A1).

3. **Mateus Komarchesqui**, Vitor Gabriel da Silva Ruffo, Luiz Fernando Carvalho, Mario Lemes Proença Jr., **eXplainable Artificial Intelligence for Transparent Optimization of Deep Learning-Based Intrusion Detection Systems**, Journal of Network and Systems Management, maio/2026, Springer, 34, 94, 10.1007/s10922-026-10084-z. (Fator de Impacto 2025: 3,6; Qualis CC 2021-2024, A2).

4. **Mateus Komarchesqui**, Vitor Gabriel da Silva Ruffo, Daniel Matheus Brandão Lent, Vinícius Ferreira Schiavon, Matheus Riki Nakanishi, Gustavo Hideyuki Kitamura Nishikawa, Luiz Fernando Carvalho, Mario Lemes Proença Jr., **A Comprehensive Survey on Concept-Drift-Resilient Network Intrusion Detection Systems**, IEEE Access, maio/2026, IEEE, 14, 69689-69718, 10.1109/ACCESS.2026.3691262. (Fator de Impacto 2025: 4,2; Qualis CC 2021-2024, A3).

Trabalhos sob revisão.

1. Alexandro Marcelo Zacaron, **Mateus Komarchesqui**, Daniel Matheus Brandão Lent, Vitor Gabriel da Silva Ruffo, Luiz Fernando Carvalho, Mario Lemes Proença Jr., Applied Soft Computing. (Fator de Impacto 2025: 7,8; Qualis CC 2021-2024, A1).

2. Vinícius Ferreira Schiavon, Vitor Gabriel da Silva Ruffo, **Mateus Komarchesqui**, Matheus Riki Nakanishi, Gustavo Hideyuki Kitamura Nishikawa, Luiz Fernando Carvalho, Mario Lemes Proença Jr., IEEE Access. (Fator de Impacto 2025: 4,2; Qualis CC 2021-2024, A3).

3. Daniel Matheus Brandão Lent, **Mateus Komarchesqui**, Jaime Lloret, Mario Lemes Proença Jr., Computer Communications. (Fator de Impacto 2025: 4,2; Qualis CC 2021-2024, A1).