UNIVERSIDADE FEDERAL DE PERNAMBUCO
CENTRO DE INFORMÁTICA
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

FABIANO CARLOS DA SILVA

HSAE: Um *Autoencoder* Híbrido Não Supervisionado para Detecção de Ataques *Zero-Day* e sua Extensão *Ensemble*

Recife

2025

FABIANO CARLOS DA SILVA

HSAE: Um *Autoencoder* Híbrido Não Supervisionado para Detecção de Ataques *Zero-Day* e sua Extensão *Ensemble*

Dissertação apresentada ao Programa de Pós Graduação em Ciência da Computação da Universi dade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação.

Área de Concentração: Redes de Computadores

Orientador: José Augusto Suruagy Monteiro

Coorientador: Rafael Roque

Recife

2025

.Catalogação de Publicação na Fonte. UFPE - Biblioteca Central

Silva, Fabiano Carlos da.
HSAE: um autoencoder híbrido não supervisionado para detecção de ataques zero-day e sua extensão ensemble / Fabiano Carlos da Silva. - Recife, 2025.
109 f.: il.

Dissertação (Mestrado) - Universidade Federal de Pernambuco, Centro de Informática, Programa de Pós-Graduação em Ciências da Computação, 2025.
Orientação: José Augusto Suruagy Monteiro.
Coorientação: Rafael Roque.
Inclui referências.

1. Detecção de anomalias; 2. Ataques zero-day; 3. Autoencoder híbrido; 4. Aprendizado não supervisionado; 5. Segurança em IoT; 6. Ensemble learning. I. Monteiro, José Augusto Suruagy. II. Roque, Rafael. III. Título.

UFPE-Biblioteca Central

Fabiano Carlos da Silva

“HSAE: Uma Abordagem Semi-Supervisionada para Detecção de Ataques Zero-Day em Tráfego de Rede”

Dissertação de mestrado apresentada ao Programa de Pós-Graduação em Ciência da Computação da Universidade Federal de Pernambuco, como requisito parcial para a obtenção do título de Mestre em Ciência da Computação. Área de Concentração: Redes de Computadores e Sistemas Distribuídos.

Aprovado em: 31/07/2025.

BANCA EXAMINADORA

_____

Prof. Dr. Djamel Fawzi Hadj Sadok
Centro de Informática / UFPE

_____

Profa. Dra. Michele Nogueira Lima
Departamento de Ciência da Computação / UFMG

_____

Prof. Dr. José Augusto Suruagy Monteiro
Centro de Informática/UFPE
(**orientador**)

Dedico este trabalho a Jesus Cristo, a quem entrego toda a honra e toda a glória. À minha mãe, por seu amor incondicional, apoio constante e presença em todos os momentos. E a todos os meus amigos, pela amizade sincera, pelo companheirismo e pela força ao longo desta jornada.

AGRADECIMENTOS

Agradeço primeiramente a Deus, fonte de toda sabedoria, força e inspiração, por ter me sustentado em cada etapa desta caminhada. Sem Ele, nada disso seria possível.

Agradeço, em segundo lugar, à minha mãe, por seu amor incondicional, dedicação, apoio e por estar sempre ao meu lado, acreditando em mim mesmo quando eu duvidava. Sua presença foi essencial em todos os momentos.

Expresso minha profunda gratidão ao meu orientador, José Augusto Suruagy, pela confiança depositada em mim, pela paciência, pelas oportunidades, pelos ensinamentos e por toda a orientação dedicada ao longo deste trabalho.

Estendo também meus sinceros agradecimentos ao coorientador Rafael Roque, pelo suporte e pelas contribuições que me ajudaram nesta pesquisa.

Um agradecimento muito especial ao meu amigo Arthur Mendes, que teve papel fundamental nesta trajetória, oferecendo orientação, ensinamentos e apoio contínuo, ajudando-me a crescer não apenas academicamente, mas também pessoalmente.

Agradeço igualmente a Paulo Freitas, pelos ensinamentos e pela disponibilidade em compartilhar conhecimento, e a todos aqueles que, de alguma forma, contribuíram para a realização deste trabalho (citados em ordem alfabética): Adrielson Pinheiro, Alex Cordeiro Cunha, David Panduro, Diego Madeira, Eduardo Arce, João Paulo Rocha, José Paulo Lima, Juliandson Ferreira, Késsia Nepomuceno, Leon Santos, Miguel Ángel Celis Carbajal, Raimundo Martins, Silas Garrido, Tairine Ferreira Pimentel, Taynara Martins, Thyago Nepomuceno e Vinícius Polito.

Ressalto que a ordem em que esses nomes aparecem não reflete o grau de importância ou o tamanho da contribuição de cada um, pois todos foram igualmente especiais e essenciais nesta caminhada. Cada ajuda, cada gesto de apoio e cada palavra de incentivo formaram elos de uma mesma corrente que tornou possível a concretização deste trabalho.

Agradeço também aos membros da banca — Michele Nogueira, Djamel Sadok, Aldri Santos e Andson Balieiro — pelas análises criteriosas, pelas observações construtivas e pelo tempo dedicado à avaliação deste trabalho, contribuindo significativamente para o seu aprimoramento.

Registro ainda minha gratidão ao projeto Mentored, por ter aberto as portas e me proporcionado oportunidades valiosas de aprendizado, colaboração e crescimento profissional, permitindo-me trabalhar junto a pessoas inspiradoras e adquirir experiências enriquecedoras.

Por fim, deixo meus sinceros agradecimentos à FACEPE e à FAPESP, pelo apoio insti-

tucional e pelo investimento na minha pesquisa, bem como a todos aqueles que, direta ou indiretamente, contribuíram nesta jornada, oferecendo apoio, incentivo e amizade ao longo do caminho.

"Security is not a product, but a process."

— Bruce Schneier

RESUMO

A detecção de ataques desconhecidos ou zero-day representa um dos principais desafios da segurança cibernética moderna, especialmente em ambientes com recursos computacionais limitados como redes IoT. Os sistemas tradicionais baseados em assinaturas são ineficazes contra ameaças desconhecidas, enquanto abordagens existentes de aprendizado de máquina apresentam limitações como dependência de parâmetros estáticos, complexidade arquitetural excessiva e necessidade de ajustes manuais constantes. Este trabalho propõe o HSAE (Hybrid Scoring Autoencoder), uma arquitetura híbrida não supervisionada que combina um autoencoder profundo com uma saída auxiliar de pontuação de anomalia, implementando detecção multi-critério através de função de perda híbrida e threshold dinâmico otimizado via Equal Error Rate (EER). Adicionalmente, desenvolve-se uma extensão ensemble que integra o HSAE, com PCA (Principal Component Analysis) e One-Class SVM (Support Vector Machine) para aumentar a robustez da detecção. A metodologia empregada baseia-se no treinamento exclusivo com dados benignos, permitindo identificação de padrões anômalos sem necessidade de exemplos de ataques previamente rotulados. A validação experimental foi conduzida nos conjuntos de dados CICIDS2017 e ToN_IoT, comparando o desempenho com o Variational Autoencoder (VAE) como modelo de referência. Os resultados demonstram superioridade consistente do HSAE, com o ensemble alcançando 94% de precisão e 96% de AUC (Area Under the Curve) na detecção de ransomware, e 96% de precisão em cenários de múltiplos ataques simultâneos. Destaca-se a redução de 99,8% no consumo de memória em relação a frameworks existentes, viabilizando implementação em dispositivos com recursos restritos. As contribuições incluem uma arquitetura adaptativa que elimina dependência de configurações manuais, metodologia transparente de pré-processamento que mitiga vieses experimentais, e validação abrangente com múltiplas métricas de desempenho. O trabalho estabelece uma solução prática e escalável para detecção proativa de ameaças em ambientes dinâmicos, equilibrando eficiência computacional com alta acurácia na identificação de ataques desconhecidos.

Palavras-chaves: Detecção de anomalias, Ataques zero-day, Autoencoder híbrido, Aprendizado não supervisionado, Segurança em IoT, Ensemble learning.

ABSTRACT

Detection of unknown or zero-day attacks represents one of the main challenges in modern cybersecurity, especially in resource-constrained environments such as IoT networks. Traditional signature-based systems are ineffective against unknown threats, while existing machine learning approaches present limitations such as dependence on static parameters, excessive architectural complexity, and the need for constant manual adjustments. This work proposes the HSAE (Hybrid Scoring Autoencoder), an unsupervised hybrid architecture that combines a deep autoencoder with an auxiliary anomaly scoring output, implementing multi-criteria detection through a hybrid loss function and dynamic threshold optimized via Equal Error Rate (EER). Additionally, an ensemble extension is developed that integrates HSAE, PCA (Principal Component Analysis), and One-Class SVM (Support Vector Machine) to enhance detection robustness. The methodology employed is based on exclusive training with benign data, enabling identification of anomalous patterns without requiring previously labeled attack examples. Experimental validation was conducted on the CICIDS2017 and ToN_IoT datasets, comparing performance with the Variational Autoencoder (VAE) as a reference model. Results demonstrate consistent superiority of HSAE, with the ensemble achieving 94% precision and 96% AUC in ransomware detection, and 96% precision in multiple simultaneous attack scenarios. Notably, a 99.8% reduction in memory consumption compared to existing frameworks was achieved, enabling implementation in resource-constrained devices. Contributions include an adaptive architecture that eliminates dependence on manual configurations, a transparent preprocessing methodology that mitigates experimental biases, and comprehensive validation with multiple performance metrics. This work establishes a practical and scalable solution for proactive threat detection in dynamic environments, balancing computational efficiency with high accuracy in identifying unknown attacks.

Keywords: Anomaly detection, Zero-day attacks, Hybrid autoencoder, Unsupervised learning, IoT security, Ensemble learning.

LISTA DE FIGURAS

Figura 1 – Cenário de Aplicação: Ambientes com Recursos Computacionais Limitados 20
Figura 2 – Taxonomia dos Principais Ataques em Redes 29
Figura 3 – Taxonomia do *Ransomware*. 36
Figura 4 – A interseção denota o comportamento que pode ser usado para construir mimicry attacks 38
Figura 5 – Estrutura do *Autoencoder* 46
Figura 6 – Exemplo de curva ROC 49
Figura 7 – Relação entre Sensibilidade, Especificidade, FPR e Threshold 50
Figura 8 – Curvas FAR e FRR em função do limiar de decisão τ. 53
Figura 9 – Arquitetura do HSAE com dupla saída 59
Figura 10 – Arquitetura do Ensemble 64
Figura 11 – Cenário de Testes para Avaliação dos Modelos 67
Figura 12 – Cenário de testes para o modelo proposto. 72
Figura 13 – Cenário de testes para o modelo ensemble. 74
Figura 14 – Comparação das curvas ROC para os modelos HSAE e o VAE usando o *dataset* CICIDS2017. 84
Figura 15 – Comparação das curvas ROC para os modelos HSAE e VAE usando o *dataset* ToN_IoT. 87
Figura 16 – Comparação das curvas ROC para o modelos *Ensemble* HSAE e *Ensemble* VAE usando o *dataset* CICIDS2017. 92
Figura 17 – Comparação das curvas ROC para os *Ensembles* usando o *dataset* ToN_IoT. 97

LISTA DE TABELAS

Tabela 1 – Limitações Identificadas versus Soluções Propostas 79
Tabela 2 – Comparação de desempenho para o conjunto de dados CICIDS2017. 81
Tabela 3 – Comparação de desempenho para o conjunto de dados ToN_IoT. 85
Tabela 4 – Comparação de desempenho dos *Ensembles* para o conjunto de dados CICIDS2017. 88
Tabela 5 – Comparação de desempenho dos *Ensembles* para o conjunto de dados ToN_IoT. 94
Tabela 6 – Comparação de Consumo de Memória por Componente 99

LISTA DE ABREVIATURAS E SIGLAS

AE	Autoencoder
AES	Advanced Encryption Standard
AUC	Area Under the Curve
CBTC	Communication-Based Train Control
CISA	Agência de Segurança Cibernética e Infraestrutura dos EUA
CNN	Convolutional Neural Network
CROSR	Classification-Reconstruction Learning for Open-Set Recognition
CSV	Comma-Separated Values
DBIR	Data Breach Investigations Report
DDoS	Distributed Denial of Service
DHRNet	Deep Hierarchical Representation Network
DID	Deep Intrusion Detection
DNS	Domain Name System
DOC	Deep Open Classification
DoS	Denial of Service
EER	Equal Error Rate
F-OSFA	Fog-based One Solution For All
FAR	False Acceptance Rate
FNR	False Negative Rate
FPIDR	False Positive Intrusion Detection Rate
FPR	False Positive Rate
FRR	False Rejection Rate
FTP	File Transfer Protocol
FTR	Fault Tolerance Rate
GMM	Gaussian Mixture Models

HSAE	Hybrid Scoring Autoencoder
HTTP	HyperText Transfer Protocol
HTTPS	HyperText Transfer Protocol Secure
IDS	Intrusion Detection System
IIoT	Industrial Internet of Things
IoMT	Internet of Medical Things
IoT	Internet of Things
IPS	Intrusion Prevention System
IRC	Internet Relay Chat
KL	Kullback-Leibler divergence
LLR	Log-Likelihood Ratio
LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MCC	Matthews Correlation Coefficient
ML	Machine Learning
NFV	Network Function Virtualization
NIST	National Institute of Standards and Technology
NRO	Network Resource Optimization
OCSVM	One-Class Support Vector Machine
PCA	Principal Component Analysis
PCAP	Packet Capture
QoS	Quality of Service
RaaS	Ransomware-as-a-Service
RAM	Random Access Memory
RBF	Radial Basis Function
RDP	Remote Desktop Protocol
RFID	Radio Frequency Identification

RSA	Rivest-Shamir-Adleman
RSS	Resident Set Size
RTID	Real-Time Intrusion Detection
SDN	Software Defined Networking
SSH	Secure Shell
TCP	Transmission Control Protocol
TCP/IP	Transmission Control Protocol/Internet Protocol
TPR	True Positive Rate
UDP	User Datagram Protocol
VAE	Variational Autoencoder
VANETs	Vehicular Ad-hoc Network
XSS	Cross-Site Scripting

LISTA DE SÍMBOLOS

μ	Média
σ	Desvio padrão
θ	Parâmetros
λ	Parâmetro de ponderação
α	Fator de ponderação
β	Fator de ponderação
τ	Limiar
ν	Parâmetro
ψ	Função encoder
φ	Função decoder

SUMÁRIO

1 INTRODUÇÃO 19
1.1 CONTEXTO 19
1.2 PROBLEMA DE PESQUISA 21
1.3 OBJETIVO 22
1.4 METODOLOGIA 23
1.5 ESTRUTURA DA DISSERTAÇÃO 24
2 REFERENCIAL TEÓRICO 25
2.1 CONCEITOS FUNDAMENTAIS SOBRE ATAQUES *ZERO-DAY* 25
2.1.1 Definição de Ataques *Zero-day* 25
2.1.2 Taxonomias e Tipos de Ataques *Zero-Day* 26
2.1.3 Desafios Específicos na Detecção 27
2.2 TAXONOMIA E COMPORTAMENTO DE ATAQUES EM REDES 27
2.2.1 Natureza dos Ataques: Ativo vs. Passivo 28
2.2.2 Classificação Geral dos Ataques em Redes 29
2.2.3 Análise Comportamental de Ataques de Negação de Serviço 29
2.2.3.1 HULK (HTTP Unbearable Load King) 30
2.2.3.2 Slowhttptest 31
2.2.3.3 Slowloris 31
2.2.3.4 GoldenEye 32
2.2.4 Ataques DDoS em IoT: Papel dos *Botnets* e *Malware* 33
2.2.5 Desafios Específicos em Ambientes IoT 33
2.2.6 Taxonomia de Ataques DDoS em IoT 34
2.2.7 *Ransomware* 35
2.3 DETECÇÃO DE ANOMALIAS EM TRÁFEGO DE REDE 36
2.4 ABORDAGENS DE DETECÇÃO 39
2.4.1 Paradigmas de Aprendizado de Máquina 40
2.4.2 Métodos Tradicionais vs. Modernos 41
2.4.3 Pré-processamento de Dados 43
2.5 TÉCNICAS NÃO SUPERVISIONADAS PARA DETECÇÃO DE ANOMALIAS 44
2.6 MÉTRICAS DE AVALIAÇÃO 47

2.6.1 Precisão, *Recall* e *F1-Score* . . . **48**
2.6.2 AUC-ROC . . . **49**
2.6.3 Métricas Específicas para Ambientes de Produção . . . **51**
2.6.4 *Equal Error Rate* (EER) . . . **52**
3 TRABALHOS RELACIONADOS . . . **54**
3.1 REVISÃO DA LITERATURA . . . 54
3.2 SÍNTESE DAS LACUNAS E REQUISITOS PARA A NOVA ABORDAGEM 56
4 ARQUITETURA PROPOSTA . . . **58**
4.1 A ARQUITETURA HSAE . . . 58
4.2 FUNDAMENTAÇÃO DA ARQUITETURA HÍBRIDA: COMBINAÇÃO ENTRE RECONSTRUÇÃO E PONTUAÇÃO DIRETA DE ANOMALIAS . . . 61
4.2.1 O Papel do *Anomaly Score* e a Função Classificatória da Saída Sigmoide . . . **61**
4.2.2 Detecção de Desvios Estruturais e Representacionais . . . **62**
4.2.3 *Score* Combinado: Uma Estratégia de Fusão de Evidências . . . **63**
4.3 *ENSEMBLE* . . . 63
4.4 DEFINIÇÃO FORMAL E MATEMÁTICA DO HSAE . . . 64
4.4.1 *Score* Combinado e Interpretação Estatística . . . **65**
4.5 DEFINIÇÃO FORMAL E MATEMÁTICA DO *ENSEMBLE* HSAE + PCA + OCSVM . . . 65
4.5.1 Teste de Hipóteses para os *Scores* . . . **66**
4.6 DEFINIÇÃO DOS EXPERIMENTOS . . . 66
4.7 IMPLEMENTAÇÃO DO CENÁRIO DE TESTES . . . 70
4.7.1 Rotulagem e Separação dos dados . . . **70**
4.7.2 Pré-processamento dos Dados . . . **71**
4.7.3 Arquitetura Base: HSAE (*Hybrid Scoring Autoencoder*) . . . **72**
4.7.4 Extensão *Ensemble*: HSAE + PCA + One-Class SVM . . . **73**
4.7.5 Modelo de Comparação: VAE (*Variational Autoencoder*) . . . **74**
4.7.6 Modelo de Comparação: VAE + PCA + One-Class SVM . . . **76**
4.7.7 Etapa de Treinamento . . . **76**
4.7.8 Síntese da Proposta e Vantagens sobre as Abordagens Correlatas . **77**
5 RESULTADOS COMPARATIVOS . . . **80**

5.1 COMPARAÇÃO DE DESEMPENHO ENTRE OS MODELOS HSAE E VAE PARA O CONJUNTO DE DADOS CICIDS2017 80
5.1.1 Análise Comparativa com ataques isolados 81
5.1.2 Visualização Comparativa dos Resultados 83
5.2 COMPARAÇÃO DE DESEMPENHO ENTRE OS MODELOS HSAE E VAE PARA O CONJUNTO DE DADOS TON_IOT 85
5.2.1 Análise Comparativa com ataques individuais 85
5.2.2 Visualização Comparativa dos Resultados 87
5.3 COMPARAÇÃO DE DESEMPENHO ENTRE OS *ENSEMBLES* NO CONJUNTO DE DADOS CICIDS2017. 88
5.3.1 Análise Comparativa Com ataques individuais 89
5.3.2 Múltiplos Ataques: Robustez em Cenários Complexos 90
5.3.3 Visualização Comparativa dos Resultados 91
5.3.4 Síntese Estratégica 93
5.4 COMPARAÇÃO DE DESEMPENHO ENTRE OS *ENSEMBLES* NO CONJUNTO DE DADOS TON_IOT. 94
5.4.1 Análise Comparativa Detalhada 94
5.4.2 Múltiplos Ataques: Robustez em Cenários Complexos 96
5.4.3 Visualização Comparativa dos Resultados 96
5.4.4 Síntese Estratégica para IoT 97
5.5 EFICIÊNCIA DE RECURSOS COMPUTACIONAIS E CONSUMO DE MEMÓRIA 98
5.5.1 Resultados Comparativos e Análise Arquitetural 98
5.5.2 Consumo Total de Sistema e Implicações Práticas 100
6 CONCLUSÃO E SUGESTÕES 102
6.1 CONCLUSÃO 102
6.2 CONTRIBUIÇÕES DA PESQUISA 103
6.3 TRABALHOS FUTUROS 103
REFERÊNCIAS 105

1 INTRODUÇÃO

1.1 CONTEXTO

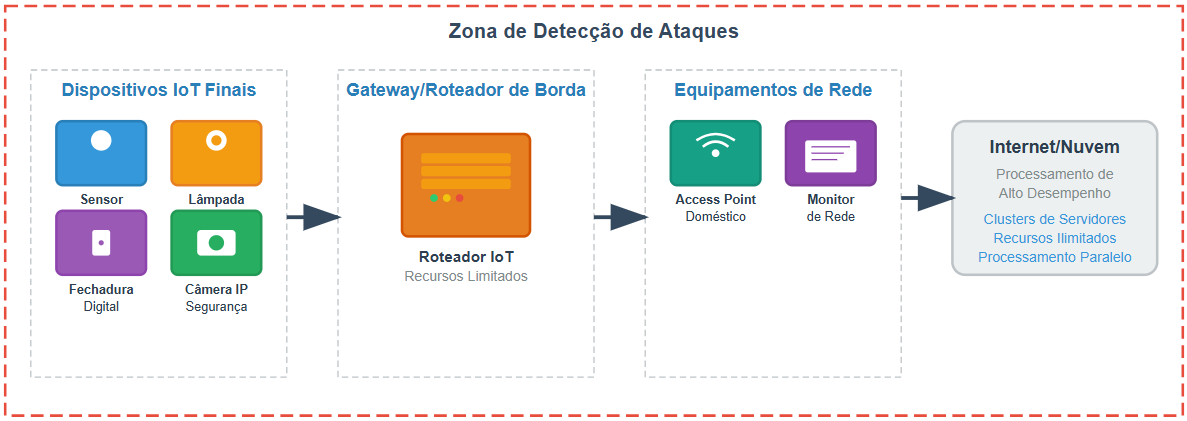
A evolução das ameaças cibernéticas revela um crescimento preocupante tanto em volume quanto em sofisticação. Um exemplo emblemático recente é o *ransomware Clop*, que em 2023 explorou vulnerabilidades do sistema MOVEit Transfer, afetando mais de 8 mil organizações globalmente, segundo o relatório da Agência de Segurança Cibernética e Infraestrutura dos EUA (CISA) citado pelo Verizon Data Breach Investigations Report (DBIR) 2024 (VERIZON, 2024). Esse tipo de ameaça, classificado como *ransomware*, um *malware* que sequestra os dados por meio de criptografia e exige pagamento para devolução, continua sendo um dos principais vetores de ataque, estando presente em 23% das violações, enquanto as técnicas de extorsão pura já representam 9%, totalizando 32% das brechas associadas a esse tipo de ameaça. Paralelamente, ataques tradicionais como o *phishing*, prática fraudulenta que visa enganar o usuário para obter dados sensíveis, mantêm sua relevância, sendo responsáveis por uma parte significativa das violações motivadas financeiramente. O relatório ainda destaca que o tempo médio para que um usuário clique em um link malicioso após abrir o e-mail de *phishing* é de apenas 21 segundos, seguido por 28 segundos para o fornecimento de dados, evidenciando a urgência de medidas preventivas (VERIZON, 2024).

A crescente conectividade no ambiente digital contemporâneo, impulsionada pela expansão da Internet das Coisas (Internet of Things (IoT)), tem ampliado significativamente a superfície de ataque em razão do número cada vez maior de dispositivos conectados. Muitos desses dispositivos operam com recursos computacionais limitados, carecem de padronização em protocolos de segurança e permanecem vulneráveis a ameaças como *spoofing* (falsificação de identidade), *jamming* (interferência no canal de comunicação) e ataques distribuídos de negação de serviço (DDoS), representando desafios críticos para a segurança da informação nesse ecossistema (LONE; MUSTAJAB; ALAM, 2023).

Neste trabalho, quando se fala em 'ambientes com recursos computacionais limitados' refere-se a dispositivos intermediários, como roteadores de borda e *gateways* de IoT, que possuem capacidade restrita de processamento e memória. Estes equipamentos são responsáveis por monitorar o tráfego de uma rede composta por diversos dispositivos finais (como sensores inteligentes e outros equipamentos IoT), mas não dispõem da alta capacidade computacional de servidores ou *clusters* de alto desempenho, tornando inviável a utilização de sistemas de

detecção mais pesados. A Figura 1 ilustra o cenário de aplicação descrito, posicionando o sistema de detecção no gateway de borda.

Figura 1 – Cenário de Aplicação: Ambientes com Recursos Computacionais Limitados



Os sistemas de detecção convencionais se baseiam predominantemente em técnicas de reconhecimento de assinaturas digitais conhecidas, apresentando limitações operacionais críticas (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023) (NEUPANE et al., 2022). A detecção por assinatura demonstra limitações significativas contra ataques polimórficos, conforme evidenciado por experimentos onde 63 produtos antivírus falharam na detecção de variantes do WannaCry (CHEN; BRIDGES, 2017). Adicionalmente, muitos sistemas baseados em *machine learning* continuam falhando ao lidar com ameaças inéditas, pois operam sob a suposição de "mundo fechado" — ou seja, assumem que os dados vistos no treinamento representam todas as possíveis entradas —, o que os torna ineficazes diante de ataques não observados previamente e conjuntos de dados desatualizados (AHMAD et al., 2023).

Entre as ameaças mais desafiadoras destacam-se os ataques *zero-day*, que exploram vulnerabilidades previamente desconhecidas pelos sistemas de defesa (BILGE; DUMITRAŞ, 2012). A literatura recente aponta que a detecção desses ataques permanece um dos grandes desafios em aberto na pesquisa em segurança cibernética (AHMAD et al., 2023) (LONE; MUSTAJAB; ALAM, 2023). Estudos indicam que essas ameaças podem persistir por longos períodos antes de serem descobertas, impactando significativamente a segurança dos sistemas. Além disso, a natureza imprevisível dos ataques *zero-day* exige o desenvolvimento de técnicas capazes de identificar padrões anômalos, mesmo diante da ausência de dados previamente rotulados, ressaltando a importância de abordagens voltadas à detecção em ambientes abertos e dinâmicos, isto é, contextos em que há constante mudança no tráfego de rede, inclusão de novos dispositivos e surgimento de vulnerabilidades desconhecidas (LONE; MUSTAJAB; ALAM, 2023).

Diante da incapacidade dos métodos tradicionais em detectar essas ameaças desconhecidas, a análise comportamental automatizada emerge como uma abordagem promissora para identificar ataques *zero-day* por meio de seus padrões anômalos. Tais técnicas visam detectar desvios comportamentais no tráfego de rede, mesmo em cenários com dados não rotulados ou ausência de assinaturas conhecidas (LONE; MUSTAJAB; ALAM, 2023). No contexto da análise de tráfego de rede, ataques *zero-day* são detectáveis através de seus desvios comportamentais, manifestando-se como padrões anômalos em relação ao tráfego legítimo (CHEN; BRIDGES, 2017). Os principais indicadores comportamentais incluem comunicação comando-e-controle através de protocolos legítimos, estabelecimento de túneis Domain Name System (DNS), padrões de tráfego modificados, anomalias temporais e conexões geográficas incomuns (VISHWAKARMA; JAIN, 2020).

O cenário evidencia uma transição em direção a abordagens mais sofisticadas, destacando a necessidade urgente de análise comportamental automatizada em tempo real e identificação proativa de anomalias (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023) (VISHWAKARMA; JAIN, 2020). Esta transição é coerente com tendências emergentes baseadas em reconhecimento de padrões anômalos e aprendizagem sem supervisão (AHMAD et al., 2023).

1.2 PROBLEMA DE PESQUISA

A detecção de ataques *zero-day* permanece um dos maiores desafios em segurança cibernética, especialmente em ecossistemas de alta complexidade e com recursos limitados, como a Internet das Coisas (IoT). Conforme discutido, sistemas de detecção convencionais, baseados em assinaturas, são ineficazes contra ameaças inéditas, e muitas abordagens de *machine learning* falham ao operar sob a premissa de "mundo fechado", tornando-se vulneráveis a ataques não vistos previamente.

Diante desse cenário, a literatura recente tem explorado diversas técnicas de aprendizado profundo não supervisionado. No entanto, uma análise crítica dos trabalhos relacionados, apresentada no Capítulo 3, revela limitações significativas que comprometem a aplicação prática dessas soluções em ambientes de produção dinâmicos. Estudos como o de (ZAVRAK; ISKEFIYELI, 2020) e (MBONA; ELOFF, 2022) propõem modelos que dependem de parâmetros estáticos, como limiares de decisão fixos, o que os torna pouco adaptáveis às variações naturais do tráfego de rede.

Outros trabalhos, embora apresentem resultados promissores, introduzem uma complexi-

dade arquitetônica excessiva, resultando em alto custo computacional e latência, fatores que inviabilizam sua implementação em dispositivos com recursos restritos (MINHAS et al., 2025) (SOLTANI et al., 2023). Além disso, a necessidade de ajustes manuais constantes e a dependência de processos complexos de recalibração (ZAHOORA et al., 2022) (SOLTANI et al., 2023) reduzem a autonomia e a robustez dos sistemas em cenários reais. Por fim, muitas propostas são validadas com um conjunto restrito de métricas, como o uso exclusivo da Area Under the Curve (AUC), o que oferece uma visão limitada de seu desempenho operacional (ZAVRAK; ISKEFIYELI, 2020).

Essas lacunas, falta de adaptabilidade, dependência de parâmetros estáticos, complexidade excessiva e avaliação incompleta evidenciam a necessidade de uma abordagem que seja, ao mesmo tempo, precisa, eficiente em termos de recursos e capaz de se generalizar para diferentes contextos de rede. Neste trabalho, o termo ataques *zero-day* refere-se especificamente a ameaças que exploram vulnerabilidades desconhecidas no tráfego de rede, com foco em cenários corporativos, IoT e IIoT. Incluem-se tanto ataques de negação de serviço (DoS/DDoS) quanto comunicações associadas a *malwares* como *ransomware*, desde que apresentem padrões comportamentais anômalos detectáveis em nível de fluxo de rede. Assim, emerge o seguinte problema de pesquisa:

Como desenvolver uma abordagem de detecção de anomalias não supervisionada eficiente e precisa para identificar ataques zero-day em dispositivos com recursos limitados, que seja generalizável para diferentes ambientes de rede?

1.3 OBJETIVO

Este trabalho visa desenvolver uma abordagem de detecção de anomalias não supervisionado que identifique ataques *zero-day* em redes com alta precisão e baixo custo computacional em diferentes tipos de ambiente de rede.

Para isso, foram estabelecidos os seguintes objetivos específicos:

- Desenvolver uma arquitetura de detecção de anomalias baseado em *autoencoder* com capacidade de identificação de ataques *zero-day*.
- Analisar o impacto de diferentes técnicas de pré-processamento e seleção de atributos no desempenho da abordagem proposta.

- Investigar a eficácia da integração de técnicas de redução de dimensionalidade e de classificação não supervisionada na melhoria da detecção de anomalias.
- Avaliar a robustez e desempenho da abordagem proposta em diferentes cenários, utilizando conjuntos de dados com perfis distintos de tráfego de rede.

1.4 METODOLOGIA

O desenvolvimento de uma solução eficaz para detecção de ataques *zero-day* demanda uma abordagem metodológica que combine pesquisa teórica, desenvolvimento de arquiteturas inovadoras e validação experimental rigorosa. Esta metodologia se propõe a guiar esse processo de forma sistemática e abrangente.

O primeiro passo consiste em realizar uma revisão sistemática do estado da arte no campo da detecção proativa de ameaças. Isso envolve a exploração de literatura acadêmica, estudos de caso, trabalhos relacionados e padronizações reconhecidas. Será conduzido um mapeamento das principais abordagens existentes, identificando suas contribuições e limitações. A compreensão dos desafios enfrentados e das práticas existentes é essencial para orientar o desenvolvimento de soluções mais eficazes e estabelecer o contexto científico da pesquisa.

Com base na revisão teórica, o próximo passo é a definição dos requisitos técnicos e funcionais que orientarão o desenvolvimento das soluções. Serão estabelecidas as restrições operacionais e os parâmetros de desempenho adequados ao contexto de aplicação. Além disso, é essencial definir os critérios de avaliação e selecionar os elementos de referência que permitirão a comparação objetiva com abordagens existentes. Os parâmetros definidos devem ser suficientemente precisos para orientar o desenvolvimento, mas flexíveis o bastante para contemplar diferentes cenários de validação.

Isso inclui o desenvolvimento e especificação detalhados das soluções propostas para o problema investigado. Por meio desta especificação será estabelecido o escopo de funcionamento, as capacidades e limitações de cada abordagem, bem como os mecanismos de adaptação para diferentes contextos de aplicação. O desenvolvimento seguirá princípios de eficiência e praticidade, considerando as restrições típicas dos ambientes de aplicação.

Uma vez estabelecida a especificação que norteia as propostas, é possível seguir para a implementação e otimização subsequente. Nesta etapa serão abordados os aspectos práticos de desenvolvimento, desde a definição das arquiteturas até a implementação dos sistemas funcio-

nais. As soluções serão desenvolvidas considerando os requisitos estabelecidos anteriormente, buscando o equilíbrio entre diferentes objetivos conflitantes identificados durante a fase de especificação.

Para demonstrar a eficácia das soluções propostas e validar suas contribuições, será desenvolvida uma estratégia de avaliação experimental abrangente. A experimentação seguirá protocolos rigorosos utilizando dados representativos e metodologias reconhecidas pela comunidade científica. Os resultados serão analisados sob múltiplas perspectivas, fornecendo evidências empíricas sobre as contribuições, limitações e aplicabilidade de cada abordagem metodológica adotada.

1.5 ESTRUTURA DA DISSERTAÇÃO

Este capítulo introdutório apresentou a contextualização do problema de pesquisa, a motivação para o estudo, seus objetivos e a metodologia geral. Os capítulos subsequentes estão estruturados para aprofundar a investigação de forma lógica e sequencial.

No Capítulo 2, é abordado o referencial teórico que serve de alicerce para a pesquisa. São explorados os conceitos de ataques de rede, os fundamentos da detecção de anomalias e as técnicas não supervisionadas, com destaque para os *autoencoders* e as métricas de avaliação de sistemas de segurança.

O Capítulo 3, por sua vez, realiza uma análise crítica dos trabalhos relacionados na literatura. Nele, são discutidas as principais abordagens existentes para a detecção de ataques *zero-day*, identificando suas limitações e as lacunas que este trabalho se propõe a preencher.

No Capítulo 4, detalha-se o método proposto. Especifica-se a arquitetura híbrida Hybrid Scoring Autoencoder (HSAE) e de sua extensão *ensemble* com Principal Component Analysis (PCA) e One-Class SVM. Adicionalmente, descreve-se toda a metodologia experimental, incluindo o tratamento dos dados, o protocolo de treinamento e os parâmetros de avaliação.

O Capítulo 5 é dedicado à apresentação e à discussão dos resultados experimentais. O desempenho das abordagens propostas é rigorosamente comparado ao de uma abordagem de referência em dois *datasets* distintos, e a eficiência computacional das soluções é avaliada em termos de consumo de memória.

Por fim, o Capítulo 6 reúne as conclusões do trabalho. Nele, destacam-se as contribuições da pesquisa para o campo da detecção de ataques *zero-day*, e são elencadas sugestões para investigações futuras.

2 REFERENCIAL TEÓRICO

A detecção proativa de ataques *zero-day* constitui elemento-chave para o desenvolvimento de sistemas de segurança cibernética eficazes, especialmente em ambientes com recursos computacionais limitados. Nesse sentido, este capítulo abordará as referências teóricas, as quais permitirão uma melhor compreensão do HSAE e da abordagem *ensemble* proposta pelo presente trabalho. Para a correta compreensão do escopo desta pesquisa, é necessário distinguir três conceitos correlacionados entre si: Segurança da Informação, Cibersegurança e Ataques em Redes. A Segurança da Informação é a disciplina mais ampla, dedicada a proteger a informação em todas as suas formas, e é historicamente fundamentada nos pilares de confidencialidade, integridade e disponibilidade — a tríade CIA (SAMONAS; COSS, 2014). A Cibersegurança, por sua vez, é frequentemente vista como uma evolução desse conceito, sendo um subconjunto da Segurança da Informação focado especificamente na proteção de ativos no ciberespaço, como redes, computadores e dados (CRAIGEN; DIAKUN-THIBAULT; PURSE, 2014). Dentro deste domínio, os Ataques em Redes são uma das principais categorias de ameaças, envolvendo ações que buscam contornar mecanismos de segurança explorando as vulnerabilidades de uma rede-alvo (HOQUE et al., 2014), tema central que será detalhado a seguir.

2.1 CONCEITOS FUNDAMENTAIS SOBRE ATAQUES *ZERO-DAY*

2.1.1 Definição de Ataques *Zero-day*

Um ataque classificado como *zero-day* ocorre quando uma vulnerabilidade é explorada antes de ser divulgada publicamente e, portanto, antes que exista uma atualização ou assinatura capaz de corrigi-la (BILGE; DUMITRAŞ, 2012). Nesse cenário, mecanismos de defesa baseados em assinaturas tornam-se ineficazes, pois a ameaça ainda não foi documentada. O termo "*zero-day*" deriva do fato de que, desde o início da exploração, não há tempo de reação disponível para desenvolvedores ou equipes de segurança (NKONGOLO; TOKMAK, 2023).

Tradicionalmente, o conceito restringia-se a vulnerabilidades inéditas. No entanto, estudos recentes indicam que ele também se aplica a vulnerabilidades conhecidas exploradas por métodos não previstos, capazes de contornar mecanismos de detecção (AHMAD et al., 2023). Essa ampliação reforça a necessidade de estabelecer limites claros entre o que é e o que não é um ataque *zero-day*. Nessa classificação, distingue-se entre *unknown unknowns* — ameaças

sem qualquer registro prévio — e *known unknowns*, que apresentam semelhanças parciais com incidentes conhecidos, mas exigem detecção adaptativa (AHMAD et al., 2023).

O impacto de ataques *zero-day* pode afetar de forma crítica a confidencialidade, integridade e disponibilidade (ROUMANI, 2021), e seu risco é ampliado no contexto de Internet das Coisas (IoT) devido à diversidade de dispositivos, heterogeneidade de protocolos e restrições de processamento e memória. Nessas redes, a exploração de vulnerabilidades ainda desconhecidas ou de vetores inéditos pode comprometer desde sensores domésticos até infraestruturas críticas, muitas vezes sem possibilidade de atualização rápida.

2.1.2 Taxonomias e Tipos de Ataques *Zero-Day*

A literatura apresenta diferentes formas de classificar ataques *zero-day*. Um critério é a origem da vulnerabilidade:

- **Vulnerabilidade nova:** falha inédita, sem registro prévio.
- **Vulnerabilidade conhecida com técnica nova:** exploração por meio de métodos inéditos capazes de evitar detecção (AHMAD et al., 2023).

Outro critério é o alvo: sistemas corporativos, dispositivos IoT, ou infraestruturas críticas — onde a exploração pode interromper serviços essenciais (NKONGOLO; TOKMAK, 2023). Também se classifica conforme o vetor de ataque, que pode envolver exploração de protocolos, serviços, aplicações ou comportamento anômalo de rede.

Há ainda casos que se destacam pelo método, como uso de código malicioso inédito, técnicas polimórficas e ofuscação para dificultar análise e detecção (GANDOTRA; BANSAL; SOFAT, 2016). Ataques a protocolos, como manipulação de pacotes e exploração de vulnerabilidades em serviços de comunicação, diferem de ataques focados em padrões comportamentais, nos quais o objetivo é imitar tráfego legítimo para evitar alarmes (*mimicry attacks*). Ataques como o Stuxnet e o WannaCry são frequentemente citados na literatura por combinarem exploração de falhas inéditas com alta capacidade de propagação (CHEN; BRIDGES, 2017). Esses casos ilustram como diferentes categorias podem se sobrepor, aumentando o desafio para a defesa.

2.1.3 Desafios Específicos na Detecção

A detecção de ataques *zero-day* enfrenta obstáculos significativos devido à dependência histórica de mecanismos baseados em assinaturas e à premissa de um "mundo fechado", onde só é identificado aquilo que já foi previamente registrado (BILGE; DUMITRAŞ, 2012). No caso de ataques inéditos, padrões anômalos no tráfego ou no comportamento do sistema tornam-se sinais precoces essenciais (AHMAD et al., 2023).

Entretanto, técnicas como *mimicry* e evasão polimórfica reduzem a eficácia de detecção. *Malwares* polimórficos, por exemplo, modificam continuamente suas assinaturas de arquivo e podem também ofuscar o tráfego de rede pós-infecção, como nas comunicações com servidores de comando e controle (CHEN; BRIDGES, 2017).

No contexto de IoT e *gateways*, esses desafios são potencializados por limitações de recursos e pela diversidade de plataformas, dificultando a aplicação de técnicas complexas de inspeção e análise em tempo real (NKONGOLO; TOKMAK, 2023). Dessa forma, a detecção efetiva exige modelos capazes de identificar anomalias comportamentais e padrões sutis de desvios, indo além da simples correspondência com ataques previamente conhecidos. Para compreender e detectar esses ataques, é necessário observar como padrões comportamentais se manifestam também em ataques já conhecidos. Embora não sejam *zero-day*, eles fornecem insights valiosos para o treinamento e validação de modelos de detecção.

Frente aos desafios expostos, este trabalho adota uma definição operacional de '*zero-day*' focada em comportamento de rede. Definimos como '*zero-day*' uma anomalia de tráfego em nível de fluxo, ausente no conjunto de treinamento, que se manifesta como um desvio do perfil de normalidade aprendido exclusivamente com dados benignos. Tal delimitação direciona nosso escopo experimental para ataques como DoS/DDoS e *ransomware*, cujos traços comportamentais na rede são o foco de nossa análise. A metodologia de avaliação completa está descrita na Seção 4.5.

2.2 TAXONOMIA E COMPORTAMENTO DE ATAQUES EM REDES

Para desenvolver um sistema capaz de detectar o comportamento de ataques desconhecidos (*zero-day*), é necessário primeiro compreender e caracterizar as 'impressões digitais comportamentais' de ataques conhecidos. A premissa central desta abordagem é que, embora a vulnerabilidade explorada por um ataque *zero-day* seja inédita, sua manifestação no tráfego

de rede frequentemente compartilhará características anômalas com famílias de ataques já estudadas. Padrões como picos de volume de tráfego, conexões de baixa taxa, exfiltração de dados ou comunicação com servidores de Comando e Controle (C&C) são exemplos de comportamentos que transcendem ataques específicos.

Portanto, esta seção se aprofunda na análise comportamental de ameaças consolidadas, como os ataques de Negação de Serviço (DoS) e o *Ransomware*. O objetivo não é criar um detector para essas ameaças específicas, mas sim utilizar seus padrões operacionais como uma base de conhecimento para treinar e validar um modelo de detecção de anomalias que seja capaz de generalizar e identificar esses mesmos tipos de desvios comportamentais quando originados por uma ameaça *zero-day*.

Os ataques em redes constituem uma das principais preocupações no cenário atual de segurança cibernética, representando ações maliciosas deliberadas que visam comprometer a confidencialidade, integridade ou disponibilidade de sistemas e dados em ambientes de comunicação digital. Segundo (HOQUE et al., 2014), "ataques em redes tentam contornar mecanismos de segurança de uma rede explorando vulnerabilidades da rede-alvo", podendo resultar em perdas financeiras significativas, vazamento de informações sensíveis e interrupção de serviços críticos.

A natureza distribuída e interconectada das redes modernas amplia consideravelmente a superfície de ataque, criando múltiplos vetores pelos quais agentes maliciosos podem penetrar nos sistemas. A crescente complexidade da infraestrutura tecnológica, aliada à proliferação de dispositivos conectados, torna o ambiente digital cada vez mais suscetível a diferentes modalidades de ataques cibernéticos. Importante salientar que a segurança da informação depende coletivamente de cada indivíduo que pode ter acesso à infraestrutura organizacional (PRABHU; THOMPSON, 2022).

2.2.1 Natureza dos Ataques: Ativo vs. Passivo

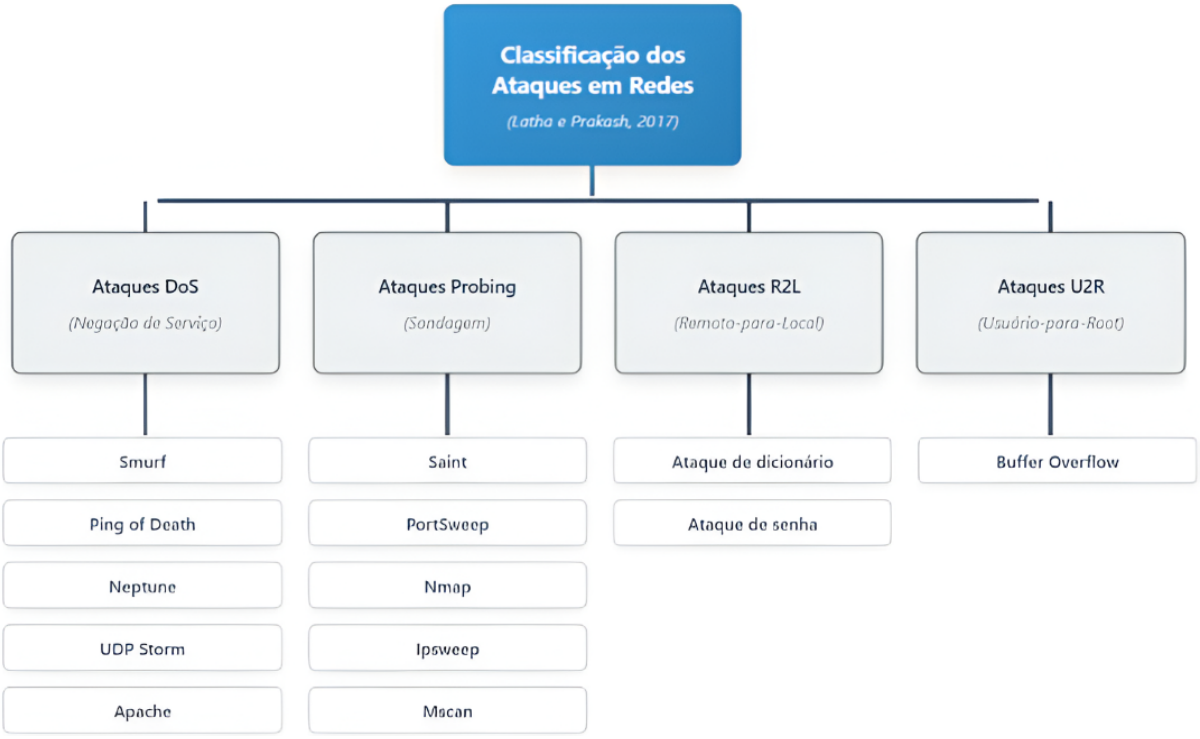
Os autores de (LATHA; PRAKASH, 2017) estabelecem uma distinção fundamental entre ataques ativos e passivos. Os ataques ativos envolvem ações que alteram recursos do sistema, como quebra de segurança ou modificação de dados. Esses ataques incluem diferentes tipos, como mascaramento (*masquerade*), repetição de sessão (*session replay*), modificação de mensagens e negação de serviço. Tais ações podem ser implementadas por meio de artefatos maliciosos como vírus, *worms*, cavalos de Tróia e inserção de código malicioso.

Ataques passivos, por outro lado, tentam conhecer ou utilizar informações importantes sem afetar os recursos do sistema. Neste tipo de ataque, o atacante utiliza ferramentas de farejamento (*sniffer*) e aguarda para capturar informações sensíveis que podem ser aplicadas em outras ações (LATHA; PRAKASH, 2017). Tais ataques incluem a liberação de conteúdo de mensagens, análise de tráfego, uso de *sniffers*, ferramentas de farejamento de pacotes e filtragem de senhas.

2.2.2 Classificação Geral dos Ataques em Redes

Latha e Prakash (2017) apresentam uma classificação estruturada onde qualquer ataque pode ser categorizado em uma das quatro categorias principais: Ataques de Negação de Serviço (DoS), Ataques de Sondagem (*Probing*), Ataques Remoto-para-Local (R2L) e Ataques Usuário-para-*Root* (U2R), conforme ilustrado na Figura 2.

Figura 2 – Taxonomia dos Principais Ataques em Redes



2.2.3 Análise Comportamental de Ataques de Negação de Serviço

Com o avanço das técnicas de ataque e a sofisticação dos atacantes, emergiram variantes específicas de ataques de negação de serviço. Embora as ameaças descritas a seguir sejam

conhecidas, a análise de seu modus operandi é fundamental para a detecção de ataques *zero-day*. Portanto, esta seção detalha ataques proeminentes como HULK, Slowhttptest, Slowloris e GoldenEye, não como um fim em si mesmos, mas como estudos de caso dos padrões comportamentais que um sistema de detecção de anomalias robusto deve ser capaz de generalizar e identificar.

1. Ataques de Negação de Serviço (DoS) Tipo de ataque onde o invasor sobrecarrega um sistema ou serviço, tornando-o indisponível para usuários legítimos. Exemplos incluem *Smurf*, *Ping of Death*, *Neptune*, *UDP Storm* e *Apache*. Uma variante específica é o ataque distribuído (DDoS), que visa comprometer a disponibilidade dos servidores inundando o canal de comunicação com solicitações falsas originadas de múltiplos dispositivos distribuídos. Segundo Neira, Kantarci e Nogueira (2023), os ataques DDoS figuram entre as principais ameaças cibernéticas globais, exigindo soluções que antecipem e mitiguem esses eventos volumétricos.

2. Ataques de Sondagem (*Probing*) Ataque onde o *hacker* escaneia uma máquina ou dispositivo de rede para descobrir seu endereço IP válido, tipo de serviço, sistema operacional utilizado e vulnerabilidades do sistema usando ferramentas de *hacking*. Essas informações podem ser usadas para explorar o sistema posteriormente. Exemplos: *Saint*, *Portsweep*, Nmap, *Ipsweep* e Mscan.

3. Ataques Remoto-para-Local (R2L) O atacante que não possui conta naquela máquina envia pacotes de rede para uma máquina vítima através da internet, estabelecendo uma conexão com aquela máquina. O atacante então causa danos ao software da máquina e pode explorar os privilégios do usuário original. Exemplos: Ataques de dicionário e Ataques de senha.

4. Ataques Usuário-para-*Root* (U2R) Um atacante se introduz na rede como usuário normal e, após atingir uma zona mais segura, tenta agir como superusuário explorando vulnerabilidades no mecanismo do computador, finalmente alcançando privilégios de superusuário. Como o atacante faz parte da rede, a identificação é muito trabalhosa. Exemplo: *buffer overflow*.

2.2.3.1 HULK (HTTP Unbearable Load King)

O ataque HULK representa um método de ataque DoS especializado em sobrecarregar servidores web através de requisições HyperText Transfer Protocol (HTTP) massivas. Esta técnica caracteriza-se por gerar um volume elevado de solicitações HTTP aparentemente legítimas, mas que consomem recursos significativos do servidor alvo (SHOREY et al., 2018). O

método HULK opera gerando URLs únicos e requisições variadas para contornar mecanismos básicos de detecção baseados em padrões, tornando cada requisição aparentemente distinta das anteriores.

A estratégia do ataque HULK fundamenta-se na exploração da capacidade limitada de processamento simultâneo de conexões dos servidores web. Ao inundar o servidor com requisições HTTP GET e POST diversificadas, este tipo de ataque esgota recursos críticos como *threads* de processamento, slots de conexão e memória disponível, resultando na indisponibilidade do serviço para usuários legítimos (MALLIGA; NANDHINI; KOGILAVANI, 2022).

2.2.3.2 Slowhttptest

O Slowhttptest constitui uma técnica de ataque desenvolvida para explorar vulnerabilidades de servidores web através de ataques DoS de baixa taxa (*slow-rate attacks*). Esta modalidade de ataque implementa diferentes variantes de ataques lentos, incluindo *Slow Headers, Slow Body e Slow Read* (SHOREY et al., 2018).

A metodologia do ataque Slowhttptest baseia-se no princípio de manter conexões HTTP abertas por períodos prolongados, enviando dados em velocidades extremamente baixas. Diferentemente dos ataques tradicionais de alta volumetria, esta abordagem explora a paciência limitada dos servidores em aguardar a conclusão de requisições aparentemente legítimas. O ataque Slow Headers, por exemplo, envia cabeçalhos HTTP de forma fragmentada e em intervalos prolongados, forçando o servidor a manter a conexão aberta enquanto aguarda a chegada completa dos dados (MALLIGA; NANDHINI; KOGILAVANI, 2022).

2.2.3.3 Slowloris

O Slowloris, desenvolvido por Robert "RSnake" Hansen, representa uma das técnicas de ataque DoS mais elegantes e eficazes já concebidas. A simplicidade desta modalidade de ataque reside no fato de que apenas um computador é necessário para derrubar um servidor web, sem afetar outras portas e serviços, impactando exclusivamente o alvo designado (SHOREY et al., 2018).

O mecanismo de funcionamento do ataque Slowloris baseia-se na abertura de numerosas conexões com o servidor web direcionado, mantendo-as abertas por período indefinido. Utilizando essas conexões, o método transmite requisições HTTP fracionárias de forma contínua,

resultando em servidores sob ataque que mantêm as conexões abertas enquanto aguardam a conclusão dessas requisições fragmentadas (SHOREY et al., 2018).

A denominação "Slowloris" deriva da característica do ataque de "lentamente" consumir os recursos HTTP do servidor. Trata-se fundamentalmente de um ataque DDoS HTTP, não devendo ser confundido com um ataque DDoS TCP. Essencialmente, o método estabelece uma conexão Transmission Control Protocol (TCP) legítima com o host alvo e, posteriormente, inunda a mesma com conexões HTTP parciais que são mantidas abertas pelo maior tempo possível e continuamente enviadas até o esgotamento completo dos recursos do alvo.

Uma vantagem significativa do Slowloris é que o atacante não envia pacotes malformados, permitindo que esses pacotes parciais atravessem facilmente sistemas de prevenção de intrusão Intrusion Prevention System (IPS) — ferramentas que, diferentemente dos sistemas que apenas detectam, são capazes de bloquear ativamente o tráfego considerado malicioso. No entanto, servidores web de gerações atuais possuem recursos adequados para mitigar ataques Slowloris através de estratégias como expansão do número máximo de clientes permitidos, restrição do número de conexões de um único endereço IP e limitação temporal para permanência de conexões (SHOREY et al., 2018).

2.2.3.4 GoldenEye

O GoldenEye constitui uma técnica de ataque DoS HTTP/S mais recente em comparação ao Slowloris. Esta modalidade de ataque, implementada originalmente em Python para fins de teste de segurança, demonstra capacidade de derrubar servidores web quando utilizada maliciosamente (SHOREY et al., 2018).

A estratégia do ataque GoldenEye consiste em utilizar o cabeçalho Connection: Keep-Alive, combinado com opções de Cache-Control, para estabelecer e manter múltiplas conexões com o servidor. Essa tática visa esgotar gradualmente todo o *pool* de *sockets* disponíveis, sufocando o servidor e impedindo que usuários legítimos consigam se conectar. O método também é descrito como uma técnica sofisticada para a análise de *malwares*, capaz de investigar ambientes de forma adaptativa para determinar as prováveis configurações do sistema-alvo (SHOREY et al., 2018).

O ataque pode alternar *online* sua condição de estrutura de sistema adaptativamente para promover investigação, sendo eficaz em descobrir qual é o ambiente almejado através de um mecanismo de execução específico para observar práticas sob situações eletivas. Embora o

GoldenEye comprometa espaço em favor da velocidade, observou-se que pode efetivamente utilizar menos espaço de memória enquanto consegue velocidade significativamente superior (SHOREY et al., 2018).

2.2.4 Ataques DDoS em IoT: Papel dos *Botnets* e *Malware*

O papel dos *botnets* IoT em ataques DDoS representa uma evolução significativa das ameaças cibernéticas. Dispositivos IoT comprometidos, conhecidos como '*bots*', são controlados por um servidor mestre (*Master Bot Controller*) que pode utilizar comunicação baseada em Internet Relay Chat (IRC), *Peer-to-Peer* ou HTTP. A formação de *botnets* IoT é facilitada pela tendência desses dispositivos permanecerem conectados 24 x 7 x 365, tornando-os alvos ideais para ataques de larga escala (VISHWAKARMA; JAIN, 2020).

Entre os *malwares* mais notórios, destaca-se o Mirai, responsável pelo maior ataque DDoS registrado até então, envolvendo até 15 milhões de dispositivos IoT com velocidade de inundação de 1 Tbps. O código-fonte do Mirai, disponibilizado publicamente, contém 62-68 pares padrão de nomes de usuário e senhas utilizados para ataques de força bruta em dispositivos IoT desprotegidos (VISHWAKARMA; JAIN, 2020).

2.2.5 Desafios Específicos em Ambientes IoT

O contexto IoT apresenta desafios únicos que amplificam as vulnerabilidades tradicionais de segurança. A Internet das Coisas emergiu como uma plataforma significativa para escalar entidades maliciosas, aproveitando-se de vulnerabilidades resultantes de recursos limitados e segurança mais fraca dos dispositivos (VISHWAKARMA; JAIN, 2020). Estes dispositivos herdam vulnerabilidades de tecnologias base como Radio Frequency Identification (RFID) e redes de sensores sem fio (DEOGIRIKAR; VIDHATE, 2017), enfrentando desafios únicos devido às limitações de processamento e recursos computacionais restritos.

A complexidade computacional dos algoritmos sofisticados torna-se inviável em ambientes IoT, especialmente em dispositivos com recursos limitados (VISHWAKARMA; JAIN, 2020). As limitações intrínsecas fazem com que as contramedidas tradicionais não possam ser aplicadas diretamente para ameaças baseadas em IoT (DEOGIRIKAR; VIDHATE, 2017). A proliferação de dispositivos IoT heterogêneos cria um ambiente particularmente propício para ataques *zero-day*, uma vez que a diversidade de sistemas operacionais, protocolos e implementações

amplia significativamente a superfície de ataque disponível para exploração de vulnerabilidades previamente desconhecidas (AHMAD et al., 2023).

Ataques DDoS IoT têm se tornado cada vez mais frequentes devido à proliferação de dispositivos IoT vulneráveis e mal configurados, especialmente em redes IoT onde o ataque visa a disponibilidade dos servidores inundando o canal de comunicação com solicitações falsas vindas de dispositivos IoT distribuídos (VISHWAKARMA; JAIN, 2020).

Essa vulnerabilidade característica dos ecossistemas IoT evidencia uma mudança de paradigma na segurança. Em redes tradicionais, a defesa é distribuída em múltiplas camadas, incluindo proteções no próprio dispositivo (*host-based*), como antivírus e *firewalls* locais, além da aplicação de *patches*. No entanto, as severas restrições de processamento e memória da maioria dos dispositivos IoT inviabilizam a implementação dessas defesas locais sofisticadas. Essa lacuna na segurança do dispositivo eleva a importância da monitoração da rede, tornando os detectores de anomalias, que analisam o tráfego de entrada e saída, uma camada de defesa essencial e, muitas vezes, a principal forma de identificar que um dispositivo foi comprometido.

2.2.6 Taxonomia de Ataques DDoS em IoT

Os ataques DDoS em redes IoT podem ser categorizados com base no impacto nas camadas da arquitetura de rede. Os ataques de camada de aplicação tentam invadir a camada de aplicação da infraestrutura de rede IoT, onde os pacotes são descartados em taxa de solicitações por segundo (Rps) devido à inundação do servidor de aplicação ou web por solicitações HTTP (Get/Post). Já os ataques de camada de infraestrutura visam tornar o sistema alvo inacessível explorando vulnerabilidades nas camadas de transporte ou de rede da arquitetura IoT, podendo ser baseados em protocolo ou volume (VISHWAKARMA; JAIN, 2020).

Estatísticas recentes mostram que ataques de camada de infraestrutura como SYN, User Datagram Protocol (UDP) e TCP *flood* obtiveram as maiores porcentagens comparados aos ataques de camada de aplicação, embora ataques HTTP *GET flood* tenham mostrado crescimento significativo. Além disso, *botnets* baseados em Linux têm se tornado mais comuns, não devido à falta de segurança do Linux, mas porque fornecedores frequentemente lançam roteadores e equipamentos IoT com *kernels* Linux desatualizados e proteção de segurança limitada (VISHWAKARMA; JAIN, 2020).

2.2.7 *Ransomware*

De forma análoga aos ataques DoS, o *ransomware*, outra ameaça crítica no cenário atual, também gera comportamentos de rede anômalos que podem ser detectados antes que o dano principal (a criptografia dos arquivos) ocorra. A análise de suas fases de comunicação com servidores de Comando e Controle (C&C) e exfiltração de dados revela padrões que, se identificados, podem sinalizar a presença de uma nova variante de *ransomware* agindo na rede. Assim, o estudo a seguir sobre a taxonomia e o funcionamento do *ransomware* serve como base para entender os desvios comportamentais que a solução proposta visa detectar.

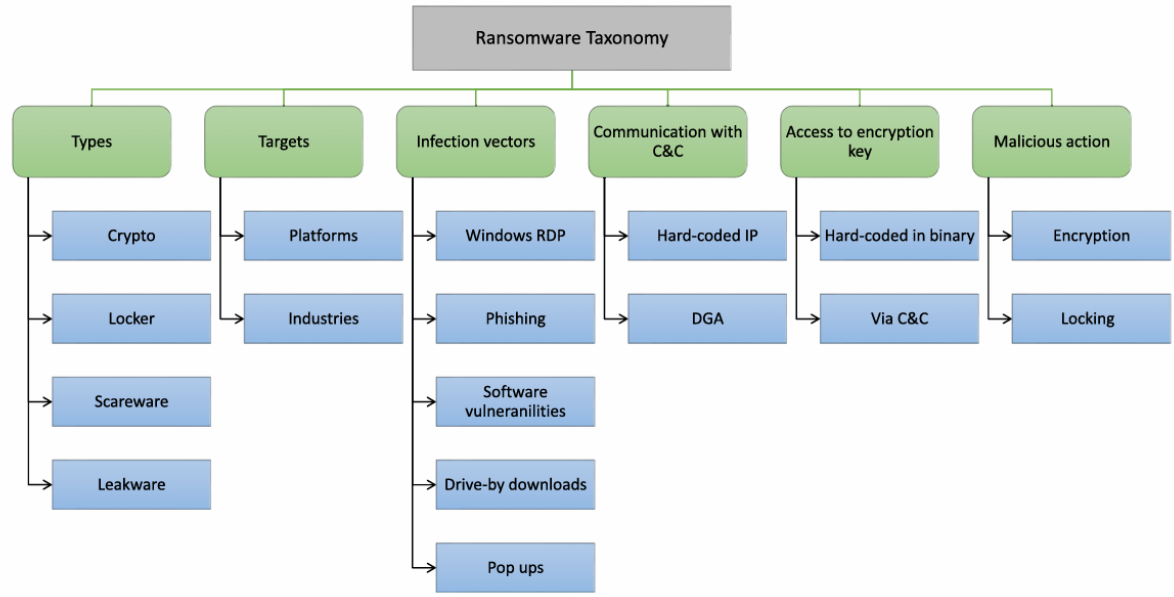
O *ransomware* representa uma das ameaças cibernéticas mais devastadoras dos últimos anos, distinguindo-se por empregar técnicas avançadas de criptografia para bloquear o acesso aos dados das vítimas e, posteriormente, exigir pagamentos para sua liberação. Essa modalidade de ataque tem experimentado um crescimento exponencial em sofisticação, impulsionado principalmente pela emergência de modelos comerciais como o Ransomware-as-a-Service (RaaS) e pela diversificação de vetores de disseminação que incluem campanhas de *phishing*, exploração de vulnerabilidades em protocolos remotos como Remote Desktop Protocol (RDP) e a utilização de kits de exploração automatizados (BEAMAN et al., 2021).

No estudo de (RAZAULLA et al., 2023), é apresentada uma taxonomia abrangente do ecossistema de *ransomware*, a qual classifica as variantes com base em seus tipos (como *crypto, locker, leakware* e *scareware*), vetores de infecção, mecanismos de comunicação com servidores de comando e controle (C&C) e ações maliciosas associadas. Essa estrutura é ilustrada na Figura 3, que integra diferentes dimensões comportamentais e técnicas, fornecendo um referencial analítico para o estudo dessa ameaça.

Essa taxonomia evidencia como diferentes famílias de *ransomware* podem compartilhar características técnicas, mesmo que variem em seus objetivos estratégicos ou métodos de disseminação. Por exemplo, o mesmo artigo analisa variantes notórias como Ryuk, REvil e Maze, e destaca como o Maze introduziu o modelo de dupla extorsão, que combina criptografia com exfiltração de dados para aumentar a pressão sobre a vítima (RAZAULLA et al., 2023).

Complementando essa perspectiva classificatória, Beaman et al. (2021) investigam os avanços mais recentes na engenharia de *ransomware*, destacando a adoção generalizada de esquemas de criptografia híbrida que combinam algoritmos Advanced Encryption Standard (AES) e Rivest-Shamir-Adleman (RSA). Os autores também documentam como eventos disruptivos globais, particularmente a pandemia de COVID-19, criaram janelas de oportunidade que foram

Figura 3 – Taxonomia do *Ransomware*.



Fonte: Razaulla et al. (2023).

amplamente exploradas por operadores maliciosos. Particularmente relevante é a demonstração de como variantes experimentais, exemplificada pelo *ransomware* AESthetic, conseguem contornar sistemas de detecção de intrusão (Intrusion Detection System (IDS)) baseados em assinatura, sublinhando a urgência no desenvolvimento de mecanismos de detecção mais sofisticados.

Em uma abordagem complementar, estudos como o de Chen e Bridges (2017) demonstram a eficácia da análise comportamental automatizada para extrair padrões de *malwares*. Essa técnica permite identificar características distintivas de variantes de *ransomware*, como o *WannaCry*, que possuem capacidades polimórficas projetadas para desafiar soluções baseadas em assinaturas estáticas. A relevância dessa abordagem é particularmente evidente na detecção precoce de ameaças, permitindo intervenções preventivas antes da execução dos processos de criptografia.

2.3 DETECÇÃO DE ANOMALIAS EM TRÁFEGO DE REDE

No contexto deste trabalho, considera-se 'anomalia' qualquer padrão de tráfego de rede que se desvia significativamente do comportamento esperado, podendo ter origem em atividades maliciosas (como ataques conhecidos e *zero-day*), falhas de configuração, erros operacionais ou eventos legítimos raros. Embora nem toda anomalia represente uma ameaça, sua detecção

é fundamental para identificar comportamentos potencialmente prejudiciais.

Diante da crescente sofisticação dos ataques cibernéticos, especialmente os ataques do tipo *zero-day* que se manifestam inicialmente como desvios comportamentais sutis no tráfego de rede, torna-se necessário compreender as abordagens de detecção de anomalias. Este tipo de detecção representa uma alternativa promissora aos métodos tradicionais baseados em assinaturas, ao permitir a identificação de ameaças ainda não catalogadas, ou mesmo variações sofisticadas de ataques já conhecidos (GARCIA-TEODORO et al., 2009).

A essência da detecção por anomalias está na identificação de padrões de comportamento que divergem de um perfil previamente estabelecido como normal. Essa abordagem envolve a modelagem estatística ou o uso de algoritmos de aprendizado de máquina treinados com dados de tráfego legítimo, a fim de detectar desvios que possam indicar atividades suspeitas. Por não depender de uma base de assinaturas previamente definida, ela se mostra particularmente eficaz contra ataques emergentes, como os *zero-day*, que exploram vulnerabilidades ainda desconhecidas pelos sistemas convencionais de defesa (HOQUE et al., 2014).

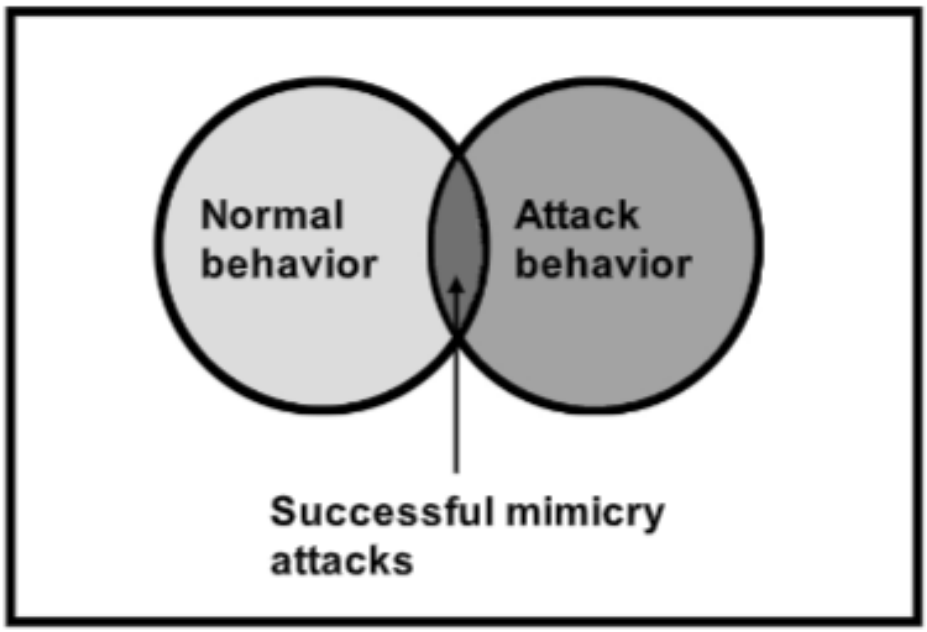
Limitações e Desafios Específicos

No entanto, nem toda atividade maliciosa resulta em anomalias perceptíveis. Um dos principais desafios enfrentados por essa abordagem é sua limitação diante de ameaças internas (*insider threats*), as quais são executadas por indivíduos com acesso legítimo diante do sistema e organização. Esses usuários, por possuírem credenciais válidas e conhecimento do funcionamento interno dos sistemas, conseguem muitas vezes operar dentro dos limites considerados normais, escapando à detecção baseada em anomalias. Liu et al. (2018) classificam essas ameaças em três categorias: traidores (usuários maliciosos), mascarados (agentes externos usando credenciais legítimas) e perpetradores não intencionais (usuários que comprometem a segurança por negligência). Reforçando essa perspectiva, Yuan e Wu (2021) afirmam que as ameaças internas são particularmente difíceis de detectar. O desafio reside no fato de que os insiders, por já possuírem acesso legítimo, não necessariamente violam controles de acesso diretos. Além disso, suas atividades maliciosas podem ser sutis, gerando comportamentos que, embora anômalos, são muito próximos aos de usuários benignos no espaço de características, dificultando a detecção por métodos estatísticos tradicionais. Para superar essa barreira, os autores defendem o uso de modelos de *Deep Learning*, como Redes Neurais Recorrentes (RNNs) e *Autoencoders*, que são capazes de aprender representações complexas e modelar sequências de comportamento para identificar esses padrões maliciosos sutis.

Além das ameaças internas, destaca-se uma classe de ataques especialmente desenhada

para contaminar sistemas de detecção comportamental: *os mimicry attacks*. Como descrevem Wagner e Soto (2002), esses ataques consistem na imitação deliberada do comportamento legítimo do sistema alvo, de forma a não acionar os mecanismos de detecção. Um exemplo prático dessa estratégia é demonstrado por Larson et al. (2009), que mostram como é possível construir longas sequências de chamadas de sistema que permanecem indetectáveis por sistemas que validam apenas os nomes das chamadas, ignorando outros contextos importantes como seus argumentos ou valores de retorno. A construção bem-sucedida desses ataques baseia-se na identificação da interseção entre comportamentos normais e maliciosos, conforme ilustrado na Figura 4, onde essa interseção representa o espaço comportamental que pode ser explorado para construir *mimicry attacks* eficazes.

Figura 4 – A interseção denota o comportamento que pode ser usado para construir mimicry attacks



Fonte: Larson et al. (2007).

O conceito de *mimicry* abrange múltiplas dimensões, tais como:

- **Mimicry temporal:** onde a temporização das ações maliciosas é ajustada para se alinhar ao ritmo normal de operação da rede;
- **Mimicry estatístico:** em que os parâmetros estatísticos do tráfego (como frequência de pacotes, tamanho de *payloads* e tempos de resposta) são manipulados para se manterem dentro dos limites normais;
- **Mimicry comportamental:** que busca replicar precisamente sequências e fluxos de uso típicos de usuários legítimos.

Para esse fim, atacantes podem empregar diversas técnicas de *fingerprinting* de tráfego de rede, que operam em diferentes níveis de detalhe, como a análise a nível de pacote (*packet-level*) ou a nível de fluxo (*flow-level*), para mapear o comportamento de sistemas e identificar alvos (SHENG et al., 2025). Um mecanismo adicional empregado é o uso de *no-ops* semânticos, instruções que não alteram o estado do sistema, mas são introduzidas para disfarçar a sequência real de ações, dificultando ainda mais a detecção (WAGNER; SOTO, 2002)). Larson et al. (2009)identificam que essas técnicas de *no-ops* podem incluir chamadas como write(-1,,0) que falham propositalmente, mas mantêm a aparência de atividade normal.

A modelagem do comportamento normal constitui, portanto, o alicerce da detecção de anomalias, exigindo análises estatísticas e algoritmos robustos capazes de lidar com variações legítimas e adaptar-se a mudanças de padrão sem comprometer a sensibilidade à ocorrência de eventos maliciosos. Larson et al. (2009) demonstram que a inclusão de informações adicionais das chamadas de sistema - como argumentos, valores de retorno e identidade do usuário - pode reduzir significativamente as opções dos atacantes para construir *mimicry attacks* bem-sucedidos, revelando manifestações de ataque previamente ocultas. Sistemas bem-sucedidos nesse campo devem ser capazes de aprender com o tráfego contínuo, manter taxas aceitáveis de falsos positivos e negativos e integrar, sempre que possível, informações contextuais e comportamentais para enriquecer sua capacidade preditiva (GARCIA-TEODORO et al., 2009) (LATHA; PRAKASH, 2017).

Portanto, embora a detecção baseada em anomalias não seja isenta de limitações, ela desempenha papel fundamental no ecossistema de defesa cibernética moderno, principalmente quando associada a abordagens híbridas e técnicas avançadas de *machine learning*. Ao possibilitar a identificação de padrões até então invisíveis a métodos tradicionais, essa estratégia torna-se indispensável para enfrentar ameaças furtivas e adaptativas. Nesse contexto, o estudo de padrões comportamentais observados em ataques conhecidos fornece subsídios valiosos para a construção de modelos capazes de reconhecer manifestações equivalentes em ataques *zero-day*, mesmo sem a existência prévia de assinaturas ou registros formais.

2.4 ABORDAGENS DE DETECÇÃO

Estabelecidos os princípios da detecção por anomalias e seus desafios intrínsecos, é necessário analisar as diferentes abordagens. A detecção de anomalias em tráfego de rede, por exemplo, evoluiu significativamente nas últimas décadas, migrando de métodos tradicionais

baseados em regras para técnicas modernas fundamentadas em aprendizado de máquina. A detecção de anomalias baseada em rede permanece como um importante campo de pesquisa e desenvolvimento em detecção de intrusão (JAVAHERI et al., 2023) (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023). Mais recentemente, a detecção de anomalias habilitada por *deep learning*, ou detecção profunda de anomalias, emergiu como uma direção relevante nesse contexto (GUO, 2023). O objetivo principal dessa abordagem, conforme destacam Pang et al. (2021), é aprender representações de características ou pontuações de anomalia por meio de redes neurais, visando identificar desvios de comportamento. Diversos métodos de detecção profunda de anomalias foram introduzidos recentemente, demonstrando desempenho superior em relação às técnicas convencionais, principalmente na resolução de problemas desafiadores e aplicações do mundo real (GUO, 2023) (BERAHMAND et al., 2024).

2.4.1 Paradigmas de Aprendizado de Máquina

As abordagens modernas para detecção de ameaças se apoiam em técnicas de Machine Learning (ML), que se mostram promissoras por sua capacidade de extrair características estatísticas de ataques (GUO, 2023). A escolha do paradigma de aprendizado é determinante e depende da disponibilidade de dados rotulados. As técnicas podem ser classificadas em três categorias principais: supervisionada, não supervisionada e semi-supervisionada (BERAHMAND et al., 2024).

- **Aprendizado Supervisionado:** Este paradigma exige um conjunto de dados de treinamento previamente rotulado, que deve conter exemplos tanto do tráfego normal quanto do anômalo (BERAHMAND et al., 2024). O objetivo do modelo é aprender a partir desses rótulos a "capturar a diferença" entre as classes (BERAHMAND et al., 2024), criando uma função que mapeia as características do tráfego a um rótulo específico (GUO, 2023). Sua principal desvantagem é a ineficácia contra ataques *zero-day*, pois, por definição, não existem exemplos rotulados de ameaças desconhecidas durante a fase de treinamento (GUO, 2023).

- **Aprendizado Não Supervisionado:** Em contraste, esta abordagem aprende padrões a partir de dados não rotulados (GUO, 2023). Na detecção de anomalias, a estratégia consiste em treinar o modelo utilizando exclusivamente dados da classe normal (GUO, 2023). O objetivo não é diferenciar classes, mas sim construir um modelo robusto do que é a "nor-

malidade", geralmente ao tentar reconstruir os dados normais com o menor erro possível. Qualquer dado que não se ajuste a este modelo e resulte em um erro de reconstrução alto é considerado uma anomalia (BERAHMAND et al., 2024). É importante notar que, dentro deste paradigma, é possível utilizar rótulos auxiliares para guiar partes específicas do treinamento, como forçar uma saída de *score* de anomalia a assumir um valor constante (e.g., zero) para todos os dados normais. Essa técnica não caracteriza o método como semi-supervisionado, pois o modelo nunca é exposto a dados da classe de anomalia durante o treinamento. Por não depender de assinaturas de ataques conhecidos, é uma abordagem inerentemente capaz de detectar anomalias novas, como os ataques *zero-day* (GUO, 2023), sendo este o foco do presente trabalho..

- **Aprendizado Semi-supervisionado:** Este paradigma utiliza uma mistura de dados no treinamento: uma pequena porção de dados rotulados e um volume maior de dados não rotulados (BERAHMAND et al., 2024). Em detecção de anomalias, isso geralmente significa ter alguns exemplos rotulados como "normal" em meio a uma grande massa de dados sem rótulos (BERAHMAND et al., 2024). O objetivo do modelo é usar os poucos dados rotulados como "âncoras" para ajudar a estruturar e a classificar o restante dos dados não rotulados, otimizando o aprendizado em cenários onde a rotulagem completa é inviável.

  Considerando o desafio de identificar ameaças desconhecidas, as técnicas não supervisionadas oferecem a flexibilidade necessária para a construção de sistemas de detecção mais robustos e independentes de assinaturas prévias.

2.4.2 Métodos Tradicionais vs. Modernos

Para uma melhor compreensão das estratégias de detecção, precisamos primeiro distinguir os Sistemas de Detecção de Intrusão (IDS) dos Sistemas de Prevenção de Intrusão (IPS). Um IDS é o processo de monitorar os eventos em um sistema ou rede, analisando-os em busca de sinais de possíveis incidentes. Sua atuação é essencialmente passiva, limitando-se a detectar atividades maliciosas e gerar alertas, sem, no entanto, alterar o tráfego de rede para bloquear a ameaça. Em contraste, um IPS possui todas as capacidades de um IDS, mas com o diferencial de poder atuar ativamente para impedir que os incidentes sejam bem-sucedidos. Ao identificar uma ameaça, o IPS pode tomar ações como finalizar sessões, bloquear conexões ou descartar pacotes, efetivamente interrompendo um ataque em andamento. A principal

diferença, portanto, é que o IDS é um sistema de monitoramento, enquanto o IPS é um sistema de controle (ABBAS; NASER; KADHIM, 2023).

No panorama histórico, as técnicas tradicionais de detecção de intrusão são predominantemente baseadas em assinaturas, regras predefinidas e limites estatísticos. Esses métodos operam comparando o tráfego de rede observado com padrões conhecidos de ataques ou comportamentos maliciosos previamente catalogados. O princípio fundamental consiste na criação de uma base de conhecimento contendo assinaturas digitais de ataques conhecidos, permitindo a identificação de ameaças através da correspondência de padrões (JAVAHERI et al., 2023) (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023) (GUO, 2023).

Referente à classificação dos Sistemas de Detecção de Intrusão (IDS), é possível agrupá-los segundo múltiplos critérios (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023):

Fonte de Dados Monitorados:

- **NIDS (Network-based IDS):** Monitoram o tráfego em pontos estratégicos da rede, sendo desafiados por questões como escalabilidade e criptografia.
- **HIDS (*Host-based* IDS):** Monitoram atividades em hosts individuais, sendo precisos para rastreamento local, mas com custos e limitações de escalabilidade.

Estratégia de Detecção:

- **Baseados em Assinatura:** Buscam padrões específicos de ataques conhecidos.
- **Baseados em Anomalia:** Detectam desvios do comportamento normal, essenciais para identificação de ataques *zero-day*, ainda que com maior taxa de falsos positivos.

Modo de Operação:

- **Tempo Real (*Online*):** Análise contínua, com consumo elevado de recursos computacionais.
- **Off-line:** Análise pós-evento, menos intensiva em recursos, porém sem resposta imediata.

Arquitetura:

- **Centralizada:** Facilita o gerenciamento, mas apresenta ponto único de falha.

- **Distribuída:** Mais escalável e resiliente.

Conforme detalhado anteriormente, o paradigma atual da detecção de anomalias aposta em abordagens de aprendizado de máquina por sua maior flexibilidade e capacidade de adaptação (GUO, 2023) (BERAHMAND et al., 2024). Dentre elas, o *deep learning* se destaca por sua aptidão para lidar com a alta dimensionalidade e a complexidade dos dados de tráfego de rede.

2.4.3 Pré-processamento de Dados

A qualidade e a representação dos dados são fatores primordiais para o sucesso na detecção de anomalias. Entre as principais técnicas de pré-processamento, destacam-se: limpeza de dados, tratamento de dados desbalanceados, conversão, seleção e extração de características, conforme detalhado a seguir.

- **Limpeza de dados:** Consiste no processo de identificar e corrigir ou remover erros, inconsistências e valores ausentes no conjunto de dados. Em dados de tráfego de rede, essa etapa é necessária para garantir que o modelo de detecção não seja treinado com informações corrompidas, o que poderia levar a uma baixa performance e a conclusões equivocadas sobre o que é um comportamento normal ou anômalo.
- **Tratamento de dados desbalanceados:** Aborda um desafio comum em segurança cibernética: a grande desproporção entre a quantidade de tráfego benigno (muito abundante) e o tráfego de ataques (eventos raros). Sem um tratamento adequado, um modelo poderia simplesmente aprender a classificar tudo como "normal", atingindo uma alta acurácia, mas sendo ineficaz para detectar ameaças. Técnicas para balancear os dados garantem que o modelo dê a devida importância aos ataques, mesmo que eles sejam minoritários.
- **Conversão de dados:** É a etapa que transforma todos os atributos do *dataset* em um formato numérico, que é o único formato que os algoritmos de aprendizado de máquina conseguem processar. Características textuais, como os nomes de protocolos de rede (ex: 'TCP', 'UDP'), precisam ser codificadas em números para que possam ser utilizadas pelo modelo durante o treinamento.
- **Seleção e extração de características:** São técnicas de redução de dimensionalidade. A seleção busca identificar e manter apenas o subconjunto de atributos mais relevantes

para a detecção, descartando os demais. Em particular, *autoencoders* e PCA são amplamente empregados para extração de características, atuando respectivamente como métodos não lineares e lineares (BERAHMAND et al., 2024).

2.5 TÉCNICAS NÃO SUPERVISIONADAS PARA DETECÇÃO DE ANOMALIAS

Uma vez estabelecidas as abordagens gerais de detecção, esta seção examina em detalhe as principais técnicas não supervisionadas, que representam a base metodológica para a identificação de ataques *zero-day* sem dependência de dados rotulados. A detecção não supervisionada em tráfego de rede compreende diferentes famílias de algoritmos, cada uma com sua própria lógica para diferenciar o comportamento normal do anômalo.

Algoritmos Baseados em Clusterização

A abordagem baseada em clusterização, como o K-Means, agrupa os dados em k *clusters* distintos com base na similaridade de suas características. O princípio para detecção de anomalias é que instâncias normais estarão próximas dos centroides (o centro) de *clusters* densos, enquanto anomalias serão pontos distantes de todos os centroides ou formarão *clusters* muito pequenos e esparsos (AHMED; SERAJ; ISLAM, 2020). Apesar de sua popularidade, o K-Means tradicional possui limitações como a necessidade de pré-definir o número de *clusters* e a sensibilidade à inicialização aleatória dos centroides, o que pode impactar seu desempenho (AHMED; SERAJ; ISLAM, 2020).

Algoritmos Baseados em Fronteira de Decisão

Técnicas como o One-Class Support Vector Machine (OCSVM) operam aprendendo uma fronteira de decisão (ou hiperplano) que envolve a maior parte dos dados de treinamento, que são considerados normais. Qualquer nova instância de dados que caia fora dessa fronteira é classificada como uma anomalia ou *outlier* (GUO, 2023). Essa abordagem é eficaz para identificar ataques que são significativamente diferentes do tráfego benigno, mas pode ter dificuldades com ataques mais complexos e sutis que se assemelham ao comportamento normal (GUO, 2023).

Algoritmos Baseados em Isolamento

Os algoritmos baseados em isolamento, como o *Isolation Forest* (iForest), partem de um princípio fundamental: anomalias são instâncias "poucas e diferentes" nos dados (FARIZI; HIDAYAH; RIZAL, 2021). Por serem raras e distintas, elas são mais fáceis de serem isoladas do que os pontos de dados normais. O método funciona construindo um conjunto de árvores de

decisão aleatórias (chamadas de *iTrees*). A lógica é que, por serem diferentes, as anomalias precisarão de menos partições para serem isoladas, resultando em um comprimento de caminho (*path length*) menor desde a raiz da árvore até o ponto ser isolado (FARIZI; HIDAYAH; RIZAL, 2021). A pontuação de anomalia de uma instância é, portanto, baseada nesse comprimento de caminho médio: quanto menor o caminho, maior a probabilidade de ser uma anomalia (FARIZI; HIDAYAH; RIZAL, 2021).

Algoritmos Baseados em Densidade

Local Outlier Factor (LOF) é um algoritmo baseado em densidade que identifica anomalias comparando a densidade local de uma instância com a de seus vizinhos. Uma instância é considerada anômala se sua densidade local for significativamente menor do que a densidade de suas vizinhanças, indicando que ela está em uma região mais esparsa do que seus vizinhos. Essa abordagem é eficaz para identificar anomalias em conjuntos de dados com densidades variadas e estruturas complexas (BUDIARTO; PERMANASARI; FAUZIATI, 2019).

Autoencoders

Autoencoders (AEs) são um tipo de rede neural artificial que aprende representações eficientes dos dados de forma não supervisionada. A arquitetura de um *autoencoder* é composta por duas partes: um encoder (codificador), que comprime os dados de entrada para uma representação de dimensão reduzida chamada de espaço latente, e um *decoder* (decodificador), que tenta reconstruir os dados de entrada originais a partir dessa representação latente (BERAHMAND et al., 2024).

O princípio fundamental para a detecção de ataques *zero-day* é treinar o *autoencoder* utilizando exclusivamente dados de tráfego normal (GUO, 2023). Ao fazer isso, o modelo se especializa em reconstruir com alta fidelidade apenas os padrões de normalidade. Quando o modelo treinado é apresentado a uma instância anômala, como um ataque desconhecido, ele falha em reconstruí-la adequadamente, gerando um erro de reconstrução elevado. Esse erro serve como uma pontuação de anomalia, e instâncias com erro acima de um determinado limiar são classificadas como maliciosas (BERAHMAND et al., 2024) (GUO, 2023). Estudos comparativos mostram que *autoencoders* geralmente superam o One-Class SVM na detecção de ataques *zero-day* complexos (GUO, 2023) (ZAVRAK; ISKEFIYELI, 2020).

Variações de *Autoencoders*:

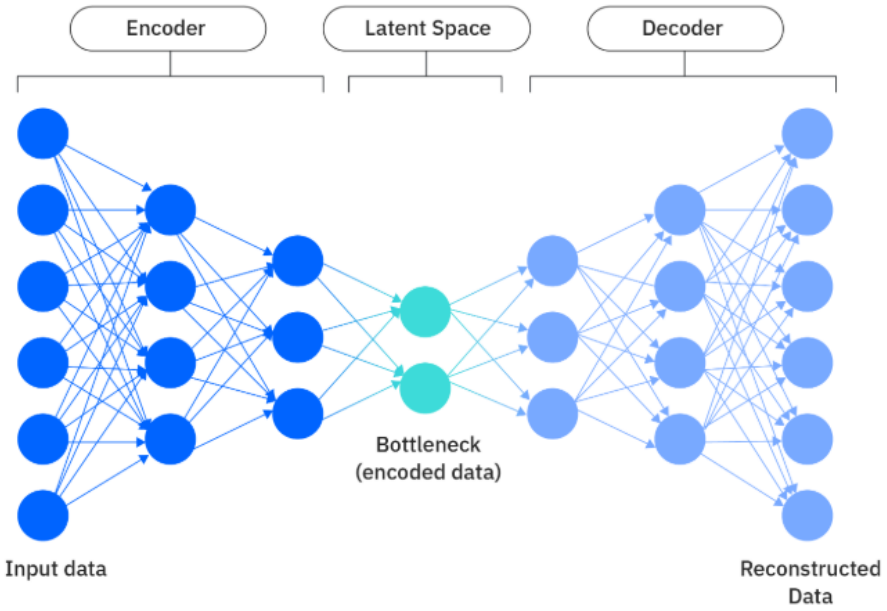
- ***Robust* AE e *Denoising* AE:** Focam na robustez em relação a ruídos.

- ***Variational* AE (VAE):** Aprendem a distribuição de probabilidade dos dados, úteis para geração e regularização do espaço latente.

Aprofundamento em *Autoencoders*

Para compreender melhor o funcionamento dos *autoencoders* na detecção de anomalias, é necessário analisar sua arquitetura e processo de treinamento em detalhes. Sua arquitetura básica compreende um encoder, um espaço latente (*bottleneck*) e um *decoder*, conforme ilustra a Figura 5. Diversos hiperparâmetros, como número de camadas, neurônios e funções de ativação, influenciam diretamente o desempenho do modelo (BERAHMAND et al., 2024).

Figura 5 – Estrutura do *Autoencoder*



Fonte: IBM (2025).

O princípio da detecção de anomalias por *autoencoders* baseia-se no erro de reconstrução. O encoder recebe o dado de entrada (como um vetor de características do tráfego de rede) e o comprime em uma representação de dimensão muito menor, chamada de espaço latente ou *bottleneck*. Essa compressão força o modelo a aprender apenas as características mais essenciais e recorrentes dos dados. Em seguida, o *decoder* recebe essa representação compacta e tem a tarefa de reconstruir o dado de entrada original com a maior fidelidade possível (BERAHMAND et al., 2024).

Quando o *autoencoder* é treinado exclusivamente com dados normais, ele se torna um especialista em comprimir e descomprimir apenas esse tipo de dado, e o espaço latente passa a ser uma representação otimizada dos padrões de normalidade (GUO, 2023). Consequentemente,

ao ser apresentado a um dado anômalo (como um ataque *zero-day*), o modelo falha em representá-lo adequadamente no espaço latente, pois nunca aprendeu os padrões daquela anomalia. Ao tentar reconstruir o dado original a partir dessa representação falha, o *decoder* produz uma saída, x', que é visivelmente diferente da entrada original, x (GUO, 2023).

Essa diferença entre a entrada e a saída é o erro de reconstrução, geralmente quantificado por uma métrica de distância como o Erro Quadrático Médio (*Mean Squared Error* - MSE), definida como $L(x, x') = ||x - x'||^2$ (GUO, 2023). Portanto, um erro de reconstrução baixo indica que o dado se conforma ao padrão de normalidade aprendido, enquanto um erro elevado é um forte indicador de uma anomalia, que pode então ser sinalizada ao exceder um limiar de detecção (BERAHMAND et al., 2024) (GUO, 2023).

Os *autoencoders* oferecem múltiplos benefícios, como redução de dimensionalidade, extração de características, compressão, remoção de ruído e detecção de anomalias. Como redutores de dimensionalidade e extratores automáticos de características, dispensam a necessidade de engenharia manual. Entretanto, enfrentam desafios como a propensão ao *overfitting*, a sensibilidade na escolha de hiperparâmetros e a possibilidade de representações enviesadas quando treinados com dados contaminados por *outliers* ou anomalias. Além disso, sua função objetivo é voltada à reconstrução e não, necessariamente, à detecção de anomalias, o que pode impactar a qualidade da representação aprendida e limitar sua eficácia Berahmand et al. (2024).

Para solucionar essas questões, aplicam-se técnicas de regularização, cujo princípio fundamental consiste em incorporar restrições na arquitetura do modelo ou em sua função de perda, com o objetivo de guiar o processo de aprendizado. Essas restrições incentivam a formação de um espaço de características mais discriminativo e com propriedades desejáveis, como impor esparsidade, aumentar a robustez a pequenas variações nos dados de entrada, ou preservar a estrutura intrínseca dos dados (BERAHMAND et al., 2024).

Pang et al. (2021) destacam que as vantagens dos métodos baseados em reconstrução incluem simplicidade e aplicabilidade geral a diferentes tipos de dados, enquanto as desvantagens envolvem o aprendizado de regularidades pouco frequentes e limitações na detecção de irregularidades raras ou complexas.

2.6 MÉTRICAS DE AVALIAÇÃO

Estabelecidas as técnicas algorítmicas fundamentais, é necessário também compreender como avaliar adequadamente o desempenho desses sistemas em cenários práticos. A avaliação

eficaz de sistemas de detecção de anomalias requer um conjunto abrangente de métricas que contemplem diferentes aspectos do desempenho e aplicabilidade prática. Estas métricas devem considerar não apenas a acurácia da detecção, mas também a viabilidade operacional em ambientes de produção.

Samariya e Thakkar (2023) enfatizam a importância de métricas específicas para detecção de anomalias, incluindo *Precision at n* (P@n), que mede a proporção de anomalias corretas nos primeiros n resultados classificados, e *Average Precision* (AP), que assume conhecimento do número total de anomalias e calcula a média das precisões em cada posição de anomalia verdadeira.

2.6.1 Precisão, *Recall* e *F1-Score*

A precisão mede a proporção de anomalias corretamente identificadas em relação ao total de detecções realizadas pelo sistema, sendo matematicamente definida como TP/(TP+FP). O *recall*, também conhecido como sensibilidade ou Taxa de Verdadeiros Positivos (TPR), quantifica a proporção de anomalias reais que foram corretamente identificadas pelo sistema, calculado como TP/(TP+FN). Conforme estabelecido por (RAINIO; TEUHO; KLÉN, 2024), estas métricas expressam respectivamente a porcentagem de instâncias corretamente classificadas no conjunto de instâncias classificadas como positivas e no conjunto de instâncias verdadeiramente positivas.

A especificidade mede a capacidade do sistema de identificar corretamente instâncias normais, sendo calculada como TN/(TN+FP). A Taxa de Falsos Positivos (FPR) relaciona-se diretamente com a especificidade através da fórmula FPR = 1 - Especificidade, conforme detalhado por (NARKHEDE, 2018) e (RAINIO; TEUHO; KLÉN, 2024).

O *F1-Score* representa a média harmônica entre precisão e *recall*, definido matematicamente como $F1 = 2 \cdot (\text{Precisão} \cdot \text{Recall})/(\text{Precisão}+\text{Recall})$. Esta métrica oferece uma avaliação balanceada que considera simultaneamente ambos os aspectos, sendo particularmente valiosa em cenários desbalanceados onde a acurácia simples pode ser enganosa (RAINIO; TEUHO; KLÉN, 2024).

2.6.2 AUC-ROC

A *Area Under the Curve - Receiver Operating Characteristic* (AUC-ROC) avalia a capacidade discriminativa geral do sistema através da análise da relação entre a taxa de verdadeiros positivos e a taxa de falsos positivos em diferentes limites de decisão. Esta métrica oferece uma visão abrangente do desempenho do classificador independentemente do limiar específico escolhido, sendo particularmente útil para comparação entre diferentes algoritmos.

A curva ROC é obtida plotando a sensibilidade (TPR - True Positive Rate) contra a taxa de falsos positivos (FPR - False Positive Rate) em todos os valores possíveis de limiar (Narkhede, 2018). Conforme descrito por (RAINIO; TEUHO; KLÉN, 2024), a curva ROC é sempre uma função monotonicamente crescente dentro do quadrado unitário ligada aos pontos (0,0) e (1,1), onde quanto mais próxima a curva ROC estiver do ponto (0,1), melhores são as predições. A figura 6 apresenta um exemplo de curva ROC (Google Developers, 2024). Narkhede (2018) destaca que a AUC-ROC representa o grau ou medida de separabilidade, indicando quanto o modelo é capaz de distinguir entre classes.

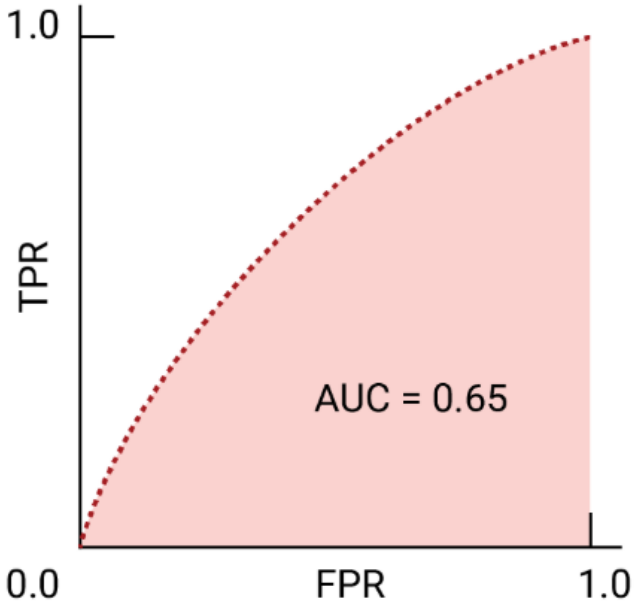


Figura 6 – Exemplo de curva ROC

A interpretação dos valores de AUC fornece insights importantes sobre o desempenho do modelo: um modelo excelente apresenta AUC próximo a 1, indicando boa medida de separabilidade; um modelo pobre apresenta AUC próximo a 0, sugerindo que está invertendo as classificações; quando AUC é 0,5, o modelo não possui capacidade de separação de classes

(NARKHEDE, 2018). Uma interpretação probabilística útil é que quando a AUC é 0,7, por exemplo, há 70% de chance de que o modelo seja capaz de distinguir corretamente entre a classe positiva e negativa (NARKHEDE, 2018).

De acordo com Narkhede (2018), existe uma relação inversa entre sensibilidade e especificidade, que é diretamente controlada pelo ajuste do limiar de decisão (*threshold*). Ao diminuir o limiar, o modelo se torna mais permissivo e classifica mais instâncias como positivas. Isso, por consequência, aumenta a sensibilidade (a capacidade de encontrar ataques verdadeiros), mas ao custo de diminuir a especificidade (pois mais tráfego normal é classificado incorretamente como ataque). De forma análoga, ao aumentar o limiar, o modelo se torna mais rigoroso, elevando a especificidade, mas reduzindo a sensibilidade. Como o FPR é calculado como 1 - especificidade, essa dinâmica implica que a Taxa de Verdadeiros Positivos (TPR, ou sensibilidade) e a FPR se movem na mesma direção: para aumentar uma, é necessário aceitar um aumento na outra. A Figura 7 ilustra a relação inversa entre sensibilidade e especificidade controlada pelo *threshold*.

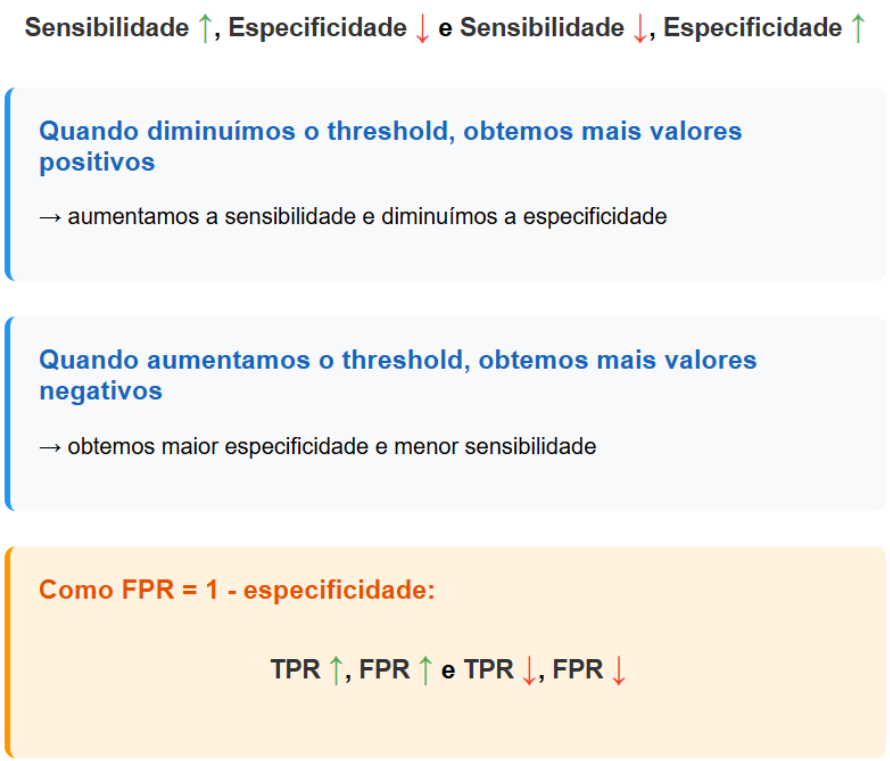


Figura 7 – Relação entre Sensibilidade, Especificidade, FPR e Threshold

Ao contrário das outras métricas, o valor da AUC não depende da escolha do limiar de decisão, tornando-se uma métrica robusta para comparação de diferentes algoritmos (RAINIO;

TEUHO; KLÉN, 2024). Para problemas de classificação multi-classe, pode-se plotar N curvas AUC-ROC para N classes usando a metodologia "Um contra Todos" (*One vs ALL*), onde cada classe é avaliada contra todas as demais combinadas (NARKHEDE, 2018).

2.6.3 Métricas Específicas para Ambientes de Produção

Taxa de Falsos Positivos e Falsos Negativos (FPR/FNR)

As taxas de falsos positivos e falsos negativos assumem importância crítica em ambientes produtivos de detecção de intrusão. A taxa de falsos positivos (FPR) mede a proporção de tráfego normal incorretamente classificado como anômalo, impactando diretamente na carga de trabalho dos analistas de segurança que precisam analisar o evento e na credibilidade do sistema. A taxa de False Negative Rate (FNR) quantifica a proporção de ataques reais que passaram despercebidos, representando um risco direto à segurança da infraestrutura.

Kumar, Selvi e Kannan (2023) enfatizam a importância de avaliar sistemas IDS em ambientes IoT com um conjunto mais abrangente de métricas. Além da tradicional Taxa de Falsos Positivos (FPR), que os autores denominam False Positive Intrusion Detection Rate (FPIDR), a pesquisa destaca métricas operacionais como Detecção de Intrusão em Tempo Real Real-Time Intrusion Detection (RTID), Taxa de Tolerância a Falhas Fault Tolerance Rate (FTR) e Otimização de Recursos de Rede Network Resource Optimization (NRO) como essenciais para validar a viabilidade prática desses sistemas em ambientes com recursos limitados.

Métricas de Qualidade de Serviço (QoS) para IoT

No contexto de IoT, Kumar, Selvi e Kannan (2023) destacam que a medição de Quality of Service (QoS) constitui uma tarefa imperativa e desafiadora. Os autores argumentam que muitos estudos utilizam apenas a taxa de falsos positivos como métrica importante, mas que uma avaliação efetiva deve incluir métricas como razão de entrega de pacotes, *delay*, energia consumida, pacotes esperados e acelerados pelos nós, e *throughput* geral da rede para medição eficaz de QoS e análise comparativa.

Consumo de Memória

O consumo de memória avalia a adequação do sistema para implementação em dispositivos com recursos limitados. Esta métrica é particularmente relevante em contextos de *edge computing* e dispositivos IoT, onde as restrições de hardware podem limitar a complexidade dos algoritmos implementáveis. A eficiência de memória também impacta a escalabilidade do sistema em ambientes com múltiplos pontos de monitoramento.

2.6.4 *Equal Error Rate* (EER)

Diante desses *trade-offs* inerentes aos sistemas de classificação, surge a necessidade de identificar um ponto operacional que equilibre adequadamente os diferentes tipos de erro. O Equal Error Rate (EER) constitui uma das principais métricas neste contexto, sendo definido como o ponto onde a False Acceptance Rate (FAR) e a False Rejection Rate (FRR) se igualam. Como destacado por (CHENG; WANG, 2004), o EER é "uma medida para avaliar o desempenho do sistema", sendo matematicamente expresso como FAR(τ^*) = FRR(τ^*), onde τ^* representa o limiar ótimo que satisfaz esta condição de igualdade.

A Figura 8 ilustra como o EER representa geometricamente a intersecção das curvas FAR e FRR quando plotadas em função do limiar de decisão, minimizando simultaneamente ambos os tipos de erro. Em sistemas baseados em modelos de Gaussian Mixture Models (GMM), é calculado através de Log-Likelihood Ratio (LLR) *scores*, onde as taxas são formalmente definidas como FAR(τ) = P(s $\geq \tau \mid H_0$) e FRR(τ) = P(s $< \tau \mid H_1$), sendo s o *score* de similaridade, H_0 a hipótese de impostor e H_1 a hipótese de usuário genuíno (CHENG; WANG, 2004).

A principal vantagem do EER é fornecer um critério objetivo para definição do limiar operacional, eliminando ajustes empíricos que dependem da intuição do desenvolvedor. Cheng e Wang (2004) demonstram que é possível estimar o EER diretamente através dos parâmetros dos modelos, sem necessidade de "um grande número de amostras de teste". Esta abordagem objetiva estabeleceu o EER como métrica padrão para benchmarking de algoritmos biométricos, sendo amplamente utilizada em competições como National Institute of Standards and Technology (NIST) *Speaker Evaluation*.

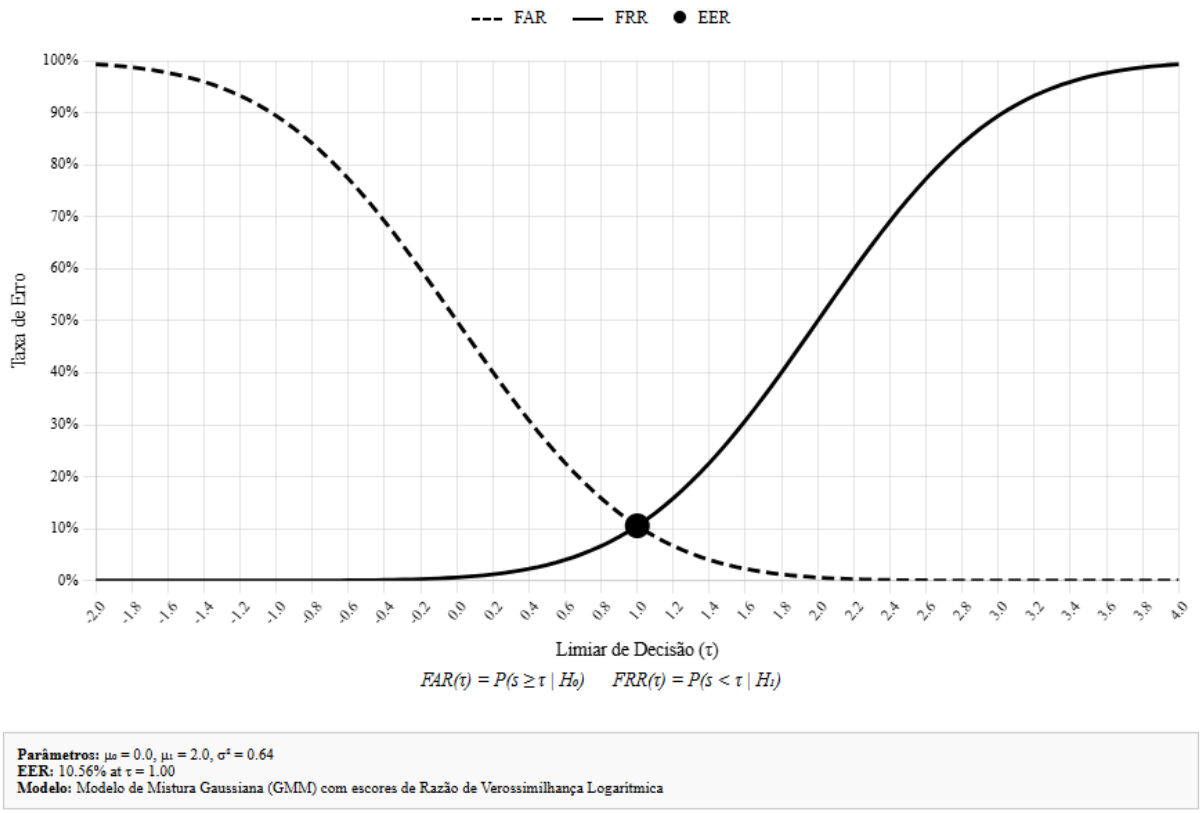


Figura 8 – Curvas FAR e FRR em função do limiar de decisão τ.

Valores menores de EER indicam melhor desempenho do sistema. Experimentos reportados em (CHENG; WANG, 2004) utilizando dados NIST 1999 demonstraram EERs de 23.7%, validando a eficácia da métrica para avaliação de sistemas reais. O EER correlaciona-se inversamente com a Área sob a Curva ROC (AUC) e representa um caso específico da Taxa de Erro Balanceada onde os custos de FAR e FRR são considerados equivalentes.

As limitações do EER incluem a dependência de distribuições representativas e a não consideração de custos assimétricos entre FAR e FRR. Como observado por Cheng e Wang (2004), "a distribuição dos *scores* computados é significativamente enviesada em relação à distribuição dos *scores* obtidos de amostras de teste", exigindo cuidado na interpretação dos resultados quando as condições operacionais diferem significativamente do ambiente de teste.

3 TRABALHOS RELACIONADOS

Este capítulo apresenta uma análise crítica dos trabalhos mais relevantes que abordam a detecção não supervisionada de ataques *zero-day*, com foco particular em ambientes de redes IoT e sistemas críticos. A análise não se limita a descrever as abordagens existentes, mas busca identificar suas principais contribuições e as lacunas e limitações que motivaram o desenvolvimento do método proposto nesta dissertação. Ao contextualizar a pesquisa frente a estudos com objetivos semelhantes, este capítulo constrói a justificativa para a arquitetura HSAE e sua extensão *ensemble*.

3.1 REVISÃO DA LITERATURA

O estudo conduzido em (ZAVRAK; ISKEFIYELI, 2020) propõe uma abordagem baseada em aprendizado profundo não supervisionado para detecção de anomalias de tráfego e ataques desconhecidos, incluindo cenários de *zero-day*, a partir de dados de fluxo. A investigação compara três métodos — Autoencoder (AE), Variational Autoencoder (VAE) e OCSVM — todos treinados exclusivamente com fluxos benignos. Os resultados experimentais indicam que o VAE apresenta desempenho superior na maioria dos cenários analisados, especialmente na detecção de ataques com alta taxa de ocorrência, como DoS e DDoS. O estudo tem sido reconhecido na literatura como uma das referências relevantes no uso de *autoencoders* para a detecção de intrusões em redes, sendo citado em revisões sistemáticas recentes que discutem o papel de modelos generativos na segurança cibernética (HALVORSEN et al., 2024) e que apresentam taxonomias atualizadas sobre sistemas de detecção de intrusões (ALKASASSBEH; BADDAR, 2023). Entretanto, os autores não exploram mecanismos adaptativos de detecção nem estratégias de controle dinâmico de falsos positivos, fatores importantes para aplicação em ambientes reais e dinâmicos. Além disso, a utilização exclusiva da métrica AUC (*Area Under the Curve*) das curvas ROC (Receiver Operating Characteristic) para avaliação limita a compreensão mais ampla do desempenho do modelo frente a diferentes aspectos do processo de detecção, como precisão, sensibilidade e taxas de erro.

Mbona e Eloff (2022) propõem detectar ataques *zero-day* combinando a Lei de Benford com aprendizado semi-supervisionado e OCSVM. Os experimentos realizados alcançam resultados razoáveis com *F1-score* de 85% e Matthews Correlation Coefficient (MCC) de 74%.

Apesar de os autores compararem diferentes métodos semi-supervisionados, o trabalho apresenta limitações para aplicação prática, utilização de *thresholds* fixos no OCSVM sem capacidade de ajuste automático às variações do tráfego, dependência de *scores* simples baseados exclusivamente na saída do classificador sem integração de múltiplas fontes de informação e ausência de mecanismos adaptativos para ambientes dinâmicos. Essas limitações, particularmente a dependência de parâmetros estáticos (*thresholds* fixos) e a falta de adaptabilidade a padrões de tráfego variáveis, comprometem a robustez do método em cenários reais onde o tráfego de rede apresenta características dinâmicas e evolutivas.

Lu et al. (2024) exploram aprendizado por transferência em sistemas Communication-Based Train Control (CBTC) usando Convolutional Neural Network (CNN) e Long Short-Term Memory (LSTM) para extrair automaticamente características espaciais e temporais. Apesar de obter resultados promissores com *F1-score* de 93,21% para *zero-day*, a necessidade constante de fine-tuning com amostras inéditas implica alta complexidade e menor eficiência computacional. Essa necessidade constante de *fine-tuning* com dados novos e a alta complexidade computacional prejudicam a aplicação em tempo real.

Minhas et al. (2025) introduzem o Fog-based One Solution For All (F-OSFA), solução *fog-based* generalizável para detecção de ataques DDoS *zero-day*, combinando CNN, árvores de decisão e *autoencoders* contrativos. Apesar da precisão elevada relatada de 96,77%, a complexidade estrutural e treinamento intensivo resultam em maior latência operacional e elevado consumo de recursos computacionais. A complexidade arquitetural excessiva e o alto consumo de recursos computacionais limitam sua aplicação prática em determinados contextos.

(ZAHOORA et al., 2022) apresentam uma abordagem baseada em *Zero-shot Learning*, método de aprendizado de máquina onde um modelo é capaz de classificar objetos ou conceitos que ele nunca viu durante o treinamento. A proposta utiliza *autoencoder* contrativo profundo e *ensemble* heterogêneo com votação, alcançando *recall* elevado de 95%. Contudo, os autores destacam a forte dependência da qualidade das representações latentes e necessidade de ajustes manuais nas regras do *ensemble*. Essa dependência crítica da qualidade das representações latentes e a necessidade de ajustes manuais nas regras do *ensemble* reduzem a robustez do método em ambientes dinâmicos.

Soltani et al. (2023) apresentam um *framework* adaptativo de quatro fases baseado em aprendizado profundo para detecção de ataques *zero-day*, integrando múltiplas implementações de *Open Set Recognition*, técnica voltada à identificação de padrões que não pertencem a nenhuma das classes previamente conhecidas, permitindo a detecção de instâncias inédi-

tas. Essa abordagem é combinada com *clustering* otimizado para classificação dinâmica de ataques conhecidos e identificação contínua de novos padrões maliciosos. O sistema incorpora uma arquitetura *end-to-end* complexa que combina reconhecimento de conjunto aberto, agrupamento inteligente, rotulagem supervisionada por grupos e atualização automática do modelo. Apesar da adaptabilidade multi-modal demonstrada, o desempenho depende criticamente da convergência inicial dos algoritmos de *clustering* e da eficácia dos múltiplos módulos integrados, podendo resultar em degradação significativa frente a ataques que mimetizam tráfego benigno, uma limitação que também afeta outras abordagens baseadas em detecção de anomalias. A dependência da qualidade inicial do *clustering*, a necessidade de re-calibração periódica dos múltiplos componentes e os custos computacionais elevados do treinamento simultâneo representam limitações para aplicação prática em ambientes de produção, exigindo expertise especializada para otimização e manutenção do *pipeline* completo.

3.2 SÍNTESE DAS LACUNAS E REQUISITOS PARA A NOVA ABORDAGEM

Os trabalhos discutidos demonstram que as técnicas atuais de detecção de ataques e anomalias apresentam limitações significativas que comprometem sua aplicação prática. As cinco limitações principais identificadas direcionam o desenvolvimento desta pesquisa: falta de mecanismos adaptativos para ambientes dinâmicos, dependência de parâmetros estáticos sem capacidade de ajuste automático, complexidade arquitetural excessiva que compromete a eficiência computacional, necessidade de ajustes manuais periódicos e dependência de intervenção humana, e avaliação limitada com uso restrito de métricas de desempenho.

A análise crítica da literatura evidencia um conjunto de desafios recorrentes que limitam a aplicação prática das soluções existentes. As principais lacunas identificadas, como a dependência de parâmetros estáticos, a complexidade arquitetural excessiva e a falta de mecanismos adaptativos, apontam para a necessidade de uma nova abordagem. Portanto, para superar essas barreiras, um sistema de detecção de ataques *zero-day* eficaz, especialmente para ambientes com recursos limitados, deve atender aos seguintes requisitos fundamentais:

- **Adaptabilidade:** Possuir mecanismos para ajustar dinamicamente seus parâmetros de detecção, como o limiar de decisão, em resposta às variações naturais do tráfego de rede.
- **Eficiência Computacional:** Apresentar uma arquitetura leve, com baixo consumo de

memória e latência, viabilizando sua implementação em dispositivos de borda e IoT.

- **Autonomia:** Reduzir a necessidade de ajustes manuais e intervenção de especialistas, automatizando o processo de calibração.
- **Avaliação Abrangente:** Ser validado por um conjunto diverso de métricas que reflitam o desempenho operacional real, para além da acurácia ou da AUC.

O método proposto no capítulo seguinte foi desenvolvido com o objetivo de satisfazer esses requisitos.

4 ARQUITETURA PROPOSTA

Este capítulo é dedicado à apresentação de uma arquitetura proposta neste trabalho. O objetivo é detalhar a arquitetura desenvolvida para superar as limitações dos *autoencoders* convencionais, discutidas no referencial teórico. Serão descritos os fundamentos teóricos que motivaram sua concepção, a estrutura do modelo e a estratégia de detecção implementada.

4.1 A ARQUITETURA HSAE

A proposta central deste estudo reside na construção de uma arquitetura de *autoencoder* denominado HSAE (*Hybrid Scoring Autoencoder*), desenvolvido para superar as limitações dos *autoencoders* tradicionais identificadas na Seção 2.4. Conforme discutido no referencial teórico, *autoencoders* convencionais dependem exclusivamente do erro de reconstrução, o que pode ser insuficiente para detectar anomalias sutis ou ataques do tipo *mimicry* (Seção 2.2). O HSAE aborda essas limitações através de uma arquitetura híbrida que combina um *autoencoder* profundo com uma ramificação auxiliar, responsável pela geração de um *score* de anomalia que emula uma probabilidade. Essa abordagem implementa uma estratégia de detecção multi-critério.

Esta arquitetura híbrida foi concebida com base em três insights teóricos fundamentais: (i) a necessidade de múltiplas perspectivas de detecção para combater ataques evasivos, conforme demonstrado na discussão sobre *mimicry attacks* (Seção 2.2.1); (ii) a importância de mecanismos adaptativos de *threshold*, evidenciada pela análise do *Equal Error Rate* (Seção 2.5.4); e (iii) as limitações inerentes de métodos baseados puramente em reconstrução, detalhadas na Seção 2.4.1. A combinação desses elementos resulta em um modelo mais robusto e adaptável.

Formalmente, seja $x \in \mathbb{R}^d$ uma amostra de entrada. O encoder mapeia x em um espaço latente $z \in \mathbb{R}^k$, onde $k < d$, por meio da função $h_\psi : \mathbb{R}^d \to \mathbb{R}^k$. O *decoder* reconstrói $\hat{x} = g_\phi(z)$, onde $g_\phi : \mathbb{R}^k \to \mathbb{R}^d$. A saída auxiliar é uma função $\hat{y} = \sigma(Wz + b)$, com σ sendo a função sigmoide.

A arquitetura completa é, portanto, uma composição de funções parametrizadas:

$$f_\theta(x) = (g_\phi(h_\psi(x)), \sigma(W h_\psi(x) + b))$$

onde $\theta = \{\psi, \phi, W, b\}$.

A Figura 9 ilustra a arquitetura completa do HSAE, destacando o fluxo de dados desde a entrada até as duas saídas: a reconstrução e a pontuação de anomalia.

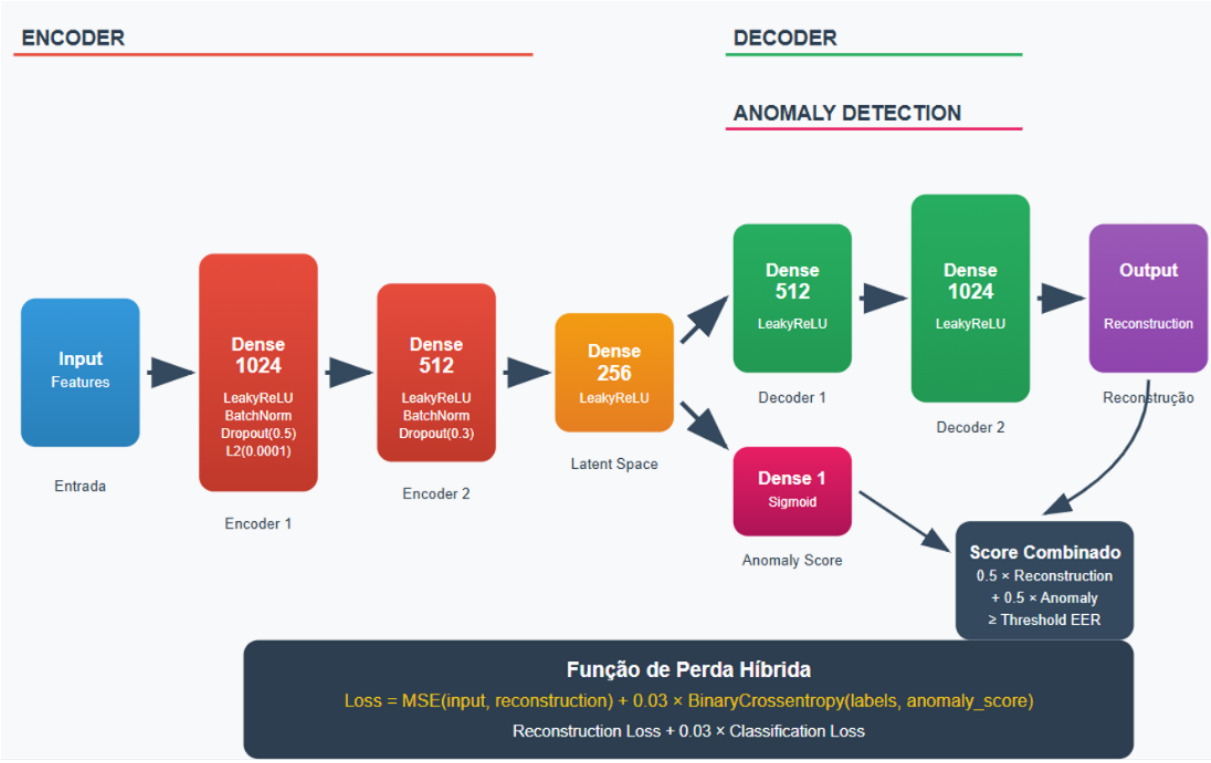


Figura 9 – Arquitetura do HSAE com dupla saída

Para implementar esta arquitetura, o HSAE é composta por três componentes principais:

- *Encoder*: *Dense*(1024) → *Dense*(512) → *Dense*(256), com *LeakyReLU*, *BatchNormalization* e *Dropout*;
- *Decoder*: *Dense*(512) → *Dense*(1024) → Output;
- Saída Auxiliar: *Dense*(1, `sigmoid`).

A concepção da arquitetura HSAE foi guiada pelo princípio de equilibrar a capacidade representacional com a eficiência computacional, evitando a "complexidade arquitetural excessiva" identificada em abordagens correlatas. A complexidade de um modelo, neste contexto, não se define apenas pelo número de camadas, mas por uma combinação de fatores que incluem a profundidade da rede, o tipo de camadas utilizadas, a aplicação de técnicas de regularização e a natureza do *pipeline* de processamento.

A estrutura utiliza uma arquitetura simétrica decrescente-crescente (1024-512-256-512-1024) baseada em compressão progressiva de informação, seguindo os princípios de redução

dimensional efetiva discutidos na Seção 2.5. Conforme definida na implementação, o encoder com três camadas densas (*Dense*) força o modelo a aprender representações cada vez mais compactas, criando um gargalo que preserva apenas as informações mais relevantes. Esta profundidade foi deliberadamente escolhida por ser suficiente para aprender as relações não-lineares complexas presentes nos dados de fluxo de rede, sem incorrer em um número excessivo de parâmetros que poderia levar ao *overfitting* e a um alto custo computacional. Essa configuração equilibra dois requisitos conflitantes: (i) capacidade representacional suficiente para evitar *underfitting* e (ii) compressão adequada para garantir sensibilidade a desvios anômalos, conforme demonstrado por (BERAHMAND et al., 2024). Adicionalmente, a escolha por camadas *Dense* é estratégica, pois são mais eficientes e adequadas para os dados tabulares (vetores de características) deste trabalho, em contraste com *frameworks* que empregam Redes Neurais Convolucionais (CNNs), arquiteturas mais pesadas e projetadas para dados com localidade espacial, como imagens.

As técnicas de regularização implementadas são importantes para este balanço, pois permitem que uma arquitetura contida generalize de forma robusta, controlando sua complexidade efetiva. Essas técnicas endereçam diretamente desafios de pré-processamento e dados desbalanceados (Seção 2.4.3). O *Dropout* (0,5 e 0,3) mitiga o risco de *overfitting* mencionado por Berahmand et al. (2024), enquanto o BatchNormalization garante estabilidade no treinamento com dados de alta dimensionalidade típicos de tráfego de rede (Seção 2.4.2). A regularização L2 (0,0001) foi calibrada para preservar a capacidade de detecção sem comprometer a sensibilidade a anomalias sutis, um *trade-off* crítico discutido na Seção 2.6.1.

Finalmente, a simplicidade do HSAE também reside em sua arquitetura unificada. O *decoder* reconstrói os dados utilizando camadas de 512 e 1024 unidades, enquanto a saída auxiliar gera uma pontuação de anomalia. Ambas as saídas são geradas a partir de um único passe pelo encoder e otimizadas conjuntamente por uma função de perda híbrida, que combina o erro de reconstrução (*Mean Squared Error* - MSE) com a entropia cruzada binária, ponderada por um fator de 0,03. Esta abordagem integrada é intrinsecamente menos complexa do que sistemas multi-estágio que acoplam modelos distintos em sequência. Para a tomada de decisão, é empregado um *score* combinado e um limiar adaptativo baseado na métrica *Equal Error Rate* (EER), tornando o mecanismo mais sensível e equilibrado. Portanto, a complexidade do HSAE foi cuidadosamente calibrada em sua profundidade, tipo de camada e estrutura geral, constituindo um modelo projetado para ser enxuto e eficaz para o problema em questão.

4.2 FUNDAMENTAÇÃO DA ARQUITETURA HÍBRIDA: COMBINAÇÃO ENTRE RECONSTRUÇÃO E PONTUAÇÃO DIRETA DE ANOMALIAS

A concepção da arquitetura HSAE foi orientada pela busca de um método de detecção de anomalias que combinasse múltiplos critérios de decisão — erro de reconstrução e *score* de classificação auxiliar —, com o objetivo de aumentar a robustez da detecção, especialmente em cenários com ataques *zero-day* e do tipo *mimicry*. A seguir, detalha-se a fundamentação para a adoção de uma saída dupla — o erro de reconstrução e o *anomaly score* — e como a combinação de ambos busca oferecer uma detecção mais robusta.

4.2.1 O Papel do *Anomaly Score* e a Função Classificatória da Saída Sigmoide

O *anomaly score* foi proposto como uma saída auxiliar que, na prática, atua como um classificador binário sobre o espaço latente — a representação comprimida e significativa dos dados gerada pelo encoder. A literatura aponta que essa representação latente pode ser utilizada como um extrator de características para outras tarefas, como a própria classificação (BANK; KOENIGSTEIN; GIRYES, 2023). Inspirado por essa capacidade, o *anomaly score* busca avaliar e classificar se a própria representação latente de uma amostra é consistente com os padrões de normalidade aprendidos.

A sua implementação utiliza uma função de ativação sigmoide, uma escolha fundamentada em suas propriedades matemáticas. Uma função de ativação sigmoide é não linear e diferenciável, requisitos para o funcionamento de redes MLP (Multilayer Perceptron) treinadas com retropropagação (*backpropagation*) (NARAYAN, 1997). Seus principais benefícios no contexto desta arquitetura são:

- **Interpretabilidade e Classificação:** A função sigmoide mapeia a saída para o intervalo [0, 1], o que permite que o resultado seja interpretado como uma pontuação de anomalia, análoga a uma probabilidade (PRATIWI et al., 2020). Essa pontuação é a base para a classificação: valores próximos de 0 são associados à classe *"normal"*, enquanto valores próximos de 1 são associados à classe *"anômala"*.
- **Treinamento Direcionado:** No treinamento com dados exclusivamente benignos, a função de perda híbrida, por meio do componente de entropia cruzada binária (Binary Cross-Entropy - BCE) — uma perda clássica de classificação —, incentiva essa saída a

se aproximar de zero (PRATIWI et al., 2020). Com isso, o modelo é treinado não apenas para reconstruir dados normais, mas também para classificar a representação latente de tráfego benigno com um *score* mínimo.

Dessa forma, a proposta é que o *anomaly score* introduza um critério de detecção classificatório e complementar, focado na consistência da representação latente dos dados.

4.2.2 Detecção de Desvios Estruturais e Representacionais

A eficácia da arquitetura híbrida é demonstrada na fase de teste, momento em que o modelo, já treinado com dados normais, confronta dados brutos que nunca viu, incluindo tanto tráfego normal quanto ataques. Ao receber uma nova amostra, o modelo a submete a duas avaliações simultâneas para decidir se é uma anomalia:

1. **Verificação Estrutural (via Reconstrução):** O modelo tenta reconstruir a amostra de entrada. O *erro de reconstrução* funciona como um sensor para **desvios estruturais**. Se uma amostra de ataque possui padrões, fluxos ou características que diferem da estrutura normal aprendida, o modelo falhará em recriá-la fielmente, gerando um erro alto.

2. **Verificação Representacional (via *Anomaly Score*):** O *encoder* mapeia a amostra para o espaço latente, e a saída sigmoide a classifica. O *anomaly score* atua como um sensor para desvios de representação. Mesmo que um ataque seja estruturalmente similar ao tráfego normal, sua representação interna no modelo pode ser atípica. A saída sigmoide, treinada para reconhecer apenas representações normais, irá sinalizar essa inconsistência com um *score* alto (próximo de 1).

A necessidade dessa dupla verificação no teste é justificada pela forma como o modelo é treinado. Se o treinamento dependesse apenas do *anomaly score*, o encoder poderia ter aprendido um atalho: colapsar a informação, mapeando todos os tipos de tráfego normal para uma representação única e simplista. Isso destruiria a capacidade do modelo de generalizar, pois ele não teria a sensibilidade para notar, no teste, as nuances que distinguem um ataque sutil de um tráfego normal. A tarefa de reconstrução, portanto, atua como um regularizador estrutural durante o treino, forçando o encoder a criar um mapa rico e detalhado da normalidade, o que

torna a verificação representacional na fase de teste muito mais confiável e precisa (BANK; KOENIGSTEIN; GIRYES, 2023).

4.2.3 *Score* Combinado: Uma Estratégia de Fusão de Evidências

A fusão dos dois *scores* em um resultado combinado busca implementar uma estratégia de combinação de informações, visando criar um indicador de anomalia mais completo. Esta abordagem encontra respaldo na literatura, onde *autoencoders* são utilizados como uma forma de regularização para redes de classificação, combinando a perda de reconstrução com a perda de classificação em uma função de custo unificada (BANK; KOENIGSTEIN; GIRYES, 2023). Abordagens similares são vistas em *autoencoders* semi-supervisionados, que também integram diferentes tipos de perdas para alavancar tanto dados rotulados quanto não rotulados (BERAHMAND et al., 2024).

A adoção de uma ponderação igualitária (50/50) representa uma abordagem inicial equilibrada, que busca assegurar que ambos os mecanismos de detecção — estrutural e representacional — contribuam de maneira balanceada para a decisão final. Essa escolha procura evitar um viés prévio em favor de um tipo específico de anomalia, contribuindo para a capacidade de generalização do modelo.

4.3 *ENSEMBLE*

Embora o modelo HSAE integre mecanismos de reconstrução e uma saída auxiliar, ainda assim pode apresentar limitações frente a certas anomalias que não geram distorções expressivas na reconstrução. Para aumentar a eficiência e reduzir falsos negativos, propomos uma arquitetura *ensemble* híbrida sequencial que combina o HSAE com técnicas de redução dimensional e classificação de *outliers*.

A abordagem *ensemble* segue uma arquitetura sequencial onde cada componente contribui com informações específicas para a detecção final, conforme ilustrado na Figura 10. O HSAE atua como extrator de características, gerando representações compactas no espaço latente e o erro de reconstrução normalizado. Sobre essas *features*, aplica-se PCA para preservar 95% da variância e eliminar ruídos. O One-Class SVM é treinado com essas *features* reduzidas para identificar anomalias com base na fronteira de normalidade. A decisão final é baseada em um *score* combinado, dado por 50% do erro de reconstrução e 50% da pontuação de anomalia

calculada pelo OCSVM. O limiar de decisão é determinado com base na métrica EER calculada sobre os dados de validação, e posteriormente aplicado aos dados de teste para a classificação final.

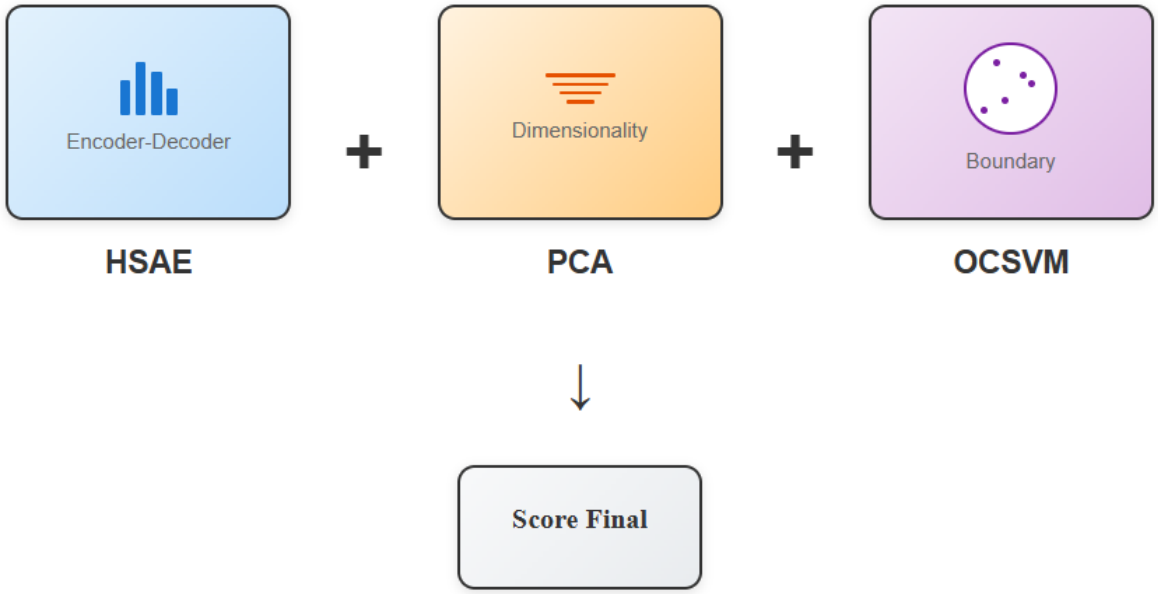


Figura 10 – Arquitetura do Ensemble

4.4 DEFINIÇÃO FORMAL E MATEMÁTICA DO HSAE

O modelo HSAE consiste em um autoencoder híbrido com duas saídas: reconstrução do vetor de entrada $\hat{x}$ e um *score* de anomalia $\hat{y}_{\text{anom}} \in [0, 1]$. O treinamento é realizado exclusivamente com dados benignos, sem a necessidade de rótulos explícitos de ataque.

Seja $x \in \mathbb{R}^n$ uma instância de tráfego de rede. O autoencoder aprende as funções:

$$f_\theta : \mathbb{R}^n \to \mathbb{R}^n \qquad \text{(reconstrução)} \tag{4.1}$$

$$s_\theta : \mathbb{R}^n \to [0, 1] \qquad \text{(score de anomalia)} \tag{4.2}$$

A função de perda total do modelo é definida como:

$$\mathcal{L}_{\text{HSAE}}(x, y) = \mathbb{E}\left[\|x - \hat{x}\|^2\right] + \lambda \cdot \text{BCE}(y, \hat{y}_{\text{anom}}) \tag{4.3}$$

onde:

- $\|x - \hat{x}\|^2$ é o erro de reconstrução;
- Rótulo auxiliar $y = 0$ para todas as instâncias benignas, de modo a forçar a saída $s_\theta(x)$ a assumir valores baixos em situações normais.
- BCE é a Binary Cross-Entropy;
- λ é o hiperparâmetro de ponderação (neste caso, $\lambda = 0,03$).

4.4.1 *Score* Combinado e Interpretação Estatística

A pontuação final atribuída a cada instância é dada por:

$$\text{score}_{\text{comb}}(x) = \alpha \cdot \text{RE}(x) + (1 - \alpha) \cdot s_\theta(x) \tag{4.4}$$

com:

- $\text{RE}(x) = \frac{1}{n} \sum_{i=1}^{n} |x_i - \hat{x}_i|$ (normalizado);
- $\alpha \in [0, 1]$ é o fator de ponderação entre reconstrução e score de anomalia.

Este *score* é interpretado como uma variável aleatória contínua $S \sim P(S|x)$, indicando a *probabilidade de anomalia*.

Interpretamos o *score* combinado como uma variável aleatória contínua S condicionada à entrada x, com densidade $p_S(s \mid x)$. Assim, $s = \text{score}_{\text{comb}}(x)$ é uma amostra da distribuição condicional de anomalia associada a x.

4.5 DEFINIÇÃO FORMAL E MATEMÁTICA DO *ENSEMBLE* HSAE + PCA + OCSVM

No ensemble híbrido, temos três estágios:

Codificador do HSAE gera vetores $z = \text{Encoder}(x)$; PCA reduz a dimensionalidade: $z' = \text{PCA}(z)$; One-Class SVM aprende uma fronteira de decisão a partir de z'.

A função de decisão do One-Class SVM é dada por:

$$h(x) = \text{sign}\left(\langle w, \phi(z') \rangle + \rho\right) \tag{4.5}$$

O *score* contínuo para cada instância é a distância ao hiperplano, invertida e normalizada:

$$s_{\text{OCSVM}}(x) = -\left(\langle w, \phi(z') \rangle + \rho\right) \tag{4.6}$$

A pontuação final do ensemble é:

$$\text{score}_{\text{ensemble}}(x) = \beta \cdot \text{RE}(x) + (1 - \beta) \cdot s_{\text{OCSVM}}(x) \tag{4.7}$$

4.5.1 Teste de Hipóteses para os *Scores*

Sejam:

- S_0: distribuição dos *scores* para $y = 0$ (benigno);
- S_1: distribuição dos *scores* para $y = 1$ (ataque).

As hipóteses estatísticas são:

$$H_0 : \mu_0 = \mu_1 \quad \text{vs} \quad H_1 : \mu_0 < \mu_1 \tag{4.8}$$

O teste t de Welch pode ser aplicado para validar a separabilidade estatística entre as distribuições.

4.6 DEFINIÇÃO DOS EXPERIMENTOS

O objetivo dos experimentos é aferir a eficácia do HSAE em sua forma básica e como parte de um *ensemble*, comparando-o com modelos representativos do estado da arte na detecção de ataques do tipo *zero-day*. Para isso, define-se um projeto experimental que avaliará os modelos em diferentes cenários operacionais e cargas de ataque, considerando anomalias de tráfego em nível de fluxo de rede. Em particular, os experimentos avaliam a capacidade do sistema, treinado exclusivamente com tráfego benigno, de: (i) detectar ataques DoS/DDoS; (ii) identificar atividades precursoras de *ransomware* antes da criptografia; (iii) reconhecer variantes e ataques *zero-day* que se manifestem como desvios do perfil benigno aprendido; e (iv) manter desempenho robusto mesmo diante de múltiplos ataques simultâneos.

Cenário de testes

O cenário de testes representa uma abordagem genérica para a avaliação dos modelos de detecção de anomalias, conforme ilustrado pela Figura 11 . O fluxo consiste em treinar os modelos exclusivamente com dados benignos e, posteriormente, avaliá-los em um ambiente com tráfego misto (benigno e malicioso).

A abordagem geral segue as etapas de:

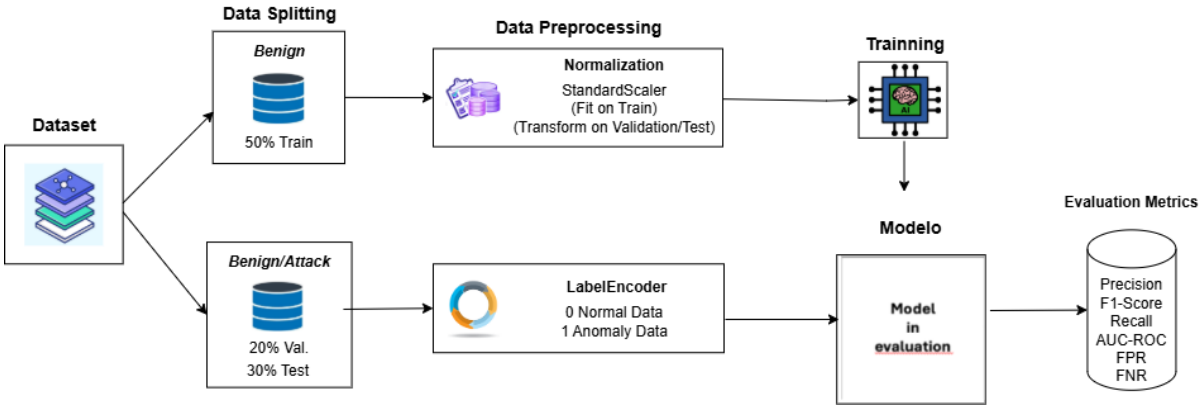


Figura 11 – Cenário de Testes para Avaliação dos Modelos

1. **Separação dos Dados (*Data Splitting*):** O tráfego benigno é dividido, com uma parte para treinamento e o restante para validação e teste.

2. **Pré-processamento (*Data Preprocessing*):** Os dados são normalizados para garantir a consistência e evitar vazamento de informações entre as etapas.

3. **Treinamento (*Training*):** Os modelos aprendem o padrão de normalidade a partir do conjunto de treino.

4. **Codificação de Rótulos (*Label Encoding*):** Transforma os rótulos textuais em valores numéricos (0 e 1) para viabilizar o cálculo das métricas de desempenho pelo modelo.

5. **Avaliação (*Evaluation*):** O desempenho dos modelos é medido em conjuntos de validação e teste, que contêm tanto dados normais quanto anômalos.

Para avaliar a eficácia da abordagem proposta, o desenho experimental foi estruturado para analisar o serviço de detecção de anomalias em tráfego de rede, especificamente em cenários de ataques *zero-day*. O desempenho do sistema é mensurado a partir de suas respostas, que podem ser classificadas em três categorias: a correta detecção de anomalias (verdadeiros positivos), a falha em detectar um ataque existente (falsos negativos) e a classificação errônea de um tráfego benigno como anômalo (falsos positivos).

A avaliação quantitativa dessa capacidade é realizada por meio de um conjunto de métricas consolidadas. A Área sob a Curva ROC (AUC) é utilizada para medir a capacidade geral do modelo em distinguir entre tráfego normal e malicioso. Para uma análise mais granular do desempenho, são empregadas a Precisão (*Precision*), que avalia a proporção de detecções

corretas dentre todos os alertas gerados, e a Revocação (*Recall*), que mede a capacidade do modelo de identificar todas as anomalias existentes. O *F1-Score* é então calculado como a média harmônica entre Precisão e Revocação, fornecendo um balanço entre ambas. Adicionalmente, a Taxa de Falsos Positivos (FPR) e a Taxa de Falsos Negativos (FNR) são inspecionadas para compreender em detalhe os tipos de erro que o modelo comete.

O núcleo do projeto experimental envolve dois fatores principais. O primeiro, um parâmetro de sistema, é o próprio modelo de detecção, que é avaliado em quatro níveis: um modelo de referência (VAE), a proposta principal (HSAE), um *ensemble* baseado no VAE (*Ensemble* VAE + One-Class SVM) e um *ensemble* baseado na proposta (*Ensemble* HSAE + PCA + One-Class SVM). O segundo, um parâmetro de carga, é o conjunto de dados, que introduz variabilidade de cenário através de dois níveis: o *dataset* CICIDS2017, representando um ambiente corporativo tradicional, e o ToN_IoT, que simula infraestruturas modernas de Internet das Coisas.

Utilizamos como primeiro conjunto de dados, o CICIDS2017 (SHARAFALDIN et al., 2018), desenvolvido pelo Canadian Institute for Cybersecurity, que simula um ambiente corporativo real, englobando tráfego legítimo e malicioso. Esse conjunto de dados é bastante utilizado em pesquisas sobre sistemas de detecção de intrusões devido à diversidade de ataques modernos que apresenta, com destaque para os ataques de negação de serviço (DoS/DDoS). Escolhemos esses ataques por sua alta frequência em redes reais e pelo impacto significativo que exercem na disponibilidade dos serviços.

Esse conjunto de dados apresenta características relevantes para a detecção de ataques *zero-day*. Composto por 2.830.540 instâncias rotuladas e 83 *features*, permite a extração de variáveis como tempo entre fluxos, estatísticas de pacotes e flags Transmission Control Protocol/Internet Protocol (TCP/IP). Conta ainda com o sistema B-Profile, que modela o comportamento de 25 usuários reais com base em protocolos como HTTP, HyperText Transfer Protocol Secure (HTTPS), File Transfer Protocol (FTP), Secure Shell (SSH) e e-mail. Além disso, 83,34% das instâncias correspondem a tráfego benigno, fornecendo um *baseline* estatisticamente robusto para identificação de desvios.

Adotamos também o conjunto de dados ToN_IoT, desenvolvido pelo Australian Centre for Cyber Security (MOUSTAFA, 2021), que representa ambientes modernos e complexos como casas inteligentes e redes industriais. Esse conjunto de dados inclui múltiplas fontes de dados, como tráfego de rede, que é processado na forma de fluxos para a extração de características detalhadas, registros de sistemas e dados de sensores, permitindo uma visão abrangente do comportamento da rede em contextos de Internet das Coisas (IoT) e IoT Industrial (Industrial

Internet of Things (IIoT)). A arquitetura que originou o ToN_IoT integra as camadas de *edge*, *fog* e *cloud*, utilizando tecnologias como Software Defined Networking (SDN), Network Function Virtualization (NFV) e Service Orchestration para coleta simultânea de eventos normais e maliciosos em ambientes realistas (MOUSTAFA, 2021).

Embora o ToN_IoT contenha uma grande variedade de vetores de ataque (*Backdoor*, *Injection*, Cross-Site Scripting (XSS), *Password Cracking*, *Man-in-the-Middle*, entre outros), foram selecionados para este trabalho apenas os ataques Denial of Service (DoS), Distributed Denial of Service (DDoS) e *Ransomware*. Incluímos o ataque de *Ransomware* devido ao seu crescimento alarmante, sendo atualmente uma das ameaças mais sofisticadas e destrutivas, conforme destacado por (RAZAULLA et al., 2023) (BEAMAN et al., 2021). Essa escolha metodológica alinha os experimentos com ameaças reais e críticas em redes operacionais modernas, conforme discutido no contexto do CICIDS2017.

Utilizar esses dois conjuntos de dados possibilita avaliar a robustez da abordagem proposta em diversos contextos operacionais e tipos de ameaças, desde ambientes corporativos tradicionais (CICIDS2017) até infraestruturas modernas e heterogêneas (ToN_IoT). Essa diversidade é essencial para validar a generalização dos modelos de detecção de intrusões, conforme recomendado na literatura recente (ELOUARDI et al., 2024). Ambos os conjuntos de dados apresentam características que estão em consonância com os critérios técnicos propostos por (GHARIB et al., 2016) para *datasets* de detecção de intrusão, favorecendo aplicações em detecção de anomalias e ameaças previamente desconhecidas.

A combinação desses fatores resulta em um total de quarenta experimentos, sendo que no *dataset* CICIDS2017 foram testados 5 ataques individuais mais ataques simultâneos, enquanto no ToN_IoT foram 3 ataques individuais mais ataques simultâneos. Na primeira etapa, a análise focou no desempenho contra ataques de forma isolada. A seleção de ameaças para esta fase visou refletir os desafios mais críticos e frequentes encontrados em redes operacionais. Do *dataset* CICIDS2017, que simula um ambiente corporativo, foram selecionados cinco tipos de ataques de Negação de Serviço: DDoS, DoS e suas variantes específicas (Slowloris, Slowhttptest, Hulk e GoldenEye). Do *dataset* ToN_IoT, representativo de ambientes modernos de IoT, foram escolhidas três categorias de ameaças: DoS, DDoS e *Ransomware*. Este teste individual de cada um dos quatro modelos contra essas ameaças permitiu criar um perfil detalhado da eficácia de cada modelo contra vetores de ataque específicos.

Na segunda etapa, o foco foi avaliar a robustez dos modelos na presença de múltiplos ataques simultâneos. O objetivo foi simular um ambiente de rede mais caótico e realista, onde

um defensor não sabe quantas ou quais ameaças estão ativas. Nessa fase, os conjuntos de teste foram compostos por uma mistura de diferentes ataques, testando a capacidade dos modelos de manter a performance e a generalização mesmo sob condições de sobrecarga e com ameaças coexistindo. Essa metodologia de duas fases visa garantir tanto uma compreensão detalhada da performance dos modelos quanto uma avaliação clara de sua aplicabilidade e resiliência em cenários de segurança complexos e atuais.

4.7 IMPLEMENTAÇÃO DO CENÁRIO DE TESTES

Esta seção detalha os procedimentos metodológicos e práticos adotados na implementação do cenário de testes. São abordados os detalhes de preparação dos dados, a configuração de implementação das arquiteturas propostas e de comparação, e os protocolos utilizados para o treinamento e a avaliação dos modelos.

4.7.1 Rotulagem e Separação dos dados

Os dados dos *datasets* selecionados foram rotulados e separados de acordo com os procedimentos a seguir. Conduzimos o processo de rotulagem por meio do algoritmo LabelEncoder, que atribui os seguintes valores:

$$\text{BENIGNO} \rightarrow 0, \quad \text{ANÔMALO} \rightarrow 1$$

Para o treinamento, foram selecionados 50% dos dados benignos, contendo exclusivamente amostras benignas. O conjunto de validação foi composto por 20% de dados benignos e a mesma quantidade de amostras maliciosas, estabelecendo o balanceamento entre as classes. O mesmo critério de balanceamento foi aplicado ao conjunto de teste, que recebeu os 30% restantes dos dados benignos e uma quantidade equivalente de ataques. A divisão dos dados foi realizada utilizando sementes de aleatoriedade fixas (`random_state`) em todas as etapas de separação. Esta metodologia opera sobre fluxos de rede, onde cada amostra individual já representa um resumo estatístico agregado de uma comunicação. As características temporais, portanto, estão encapsuladas nos atributos de cada fluxo, tornando a divisão por amostras uma abordagem consistente com a prática padrão da literatura para *datasets* desta natureza (JAVAHERI et al., 2023) (ABDULGANIYU; TCHAKOUCHT; SAHEED, 2023). O conjunto

de treinamento $\mathcal{D}$_train é definido como:

$$\mathcal{D}_{\text{train}} = \{(x_i, y_i) \in \mathcal{X}_{\text{train}} \times \mathcal{Y} \mid y_i = 0\}$$

ou seja, é composto exclusivamente por amostras benignas, permitindo ao modelo, neste caso, o *autoencoder* aprender a estrutura normal do tráfego de rede em ausência de ruído malicioso. Os conjuntos de validação e teste são definidos como:

$$\mathcal{D}_{\text{val}} = \{(x_i, y_i) \in \mathcal{X}_{\text{val}} \times \mathcal{Y} \mid y_i \in \{0, 1\} \text{ e } N_0 = N_1\}$$

$$\mathcal{D}_{\text{test}} = \{(x_i, y_i) \in \mathcal{X}_{\text{test}} \times \mathcal{Y} \mid y_i \in \{0, 1\} \text{ e } N_0 = N_1\}$$

onde N_0 e N_1 representam a cardinalidade dos subconjuntos de instâncias benignas e maliciosas, respectivamente, garantindo o balanceamento entre as classes no conjunto de teste. Essa divisão estratégica permite avaliar a capacidade de generalização do modelo frente a dados não vistos, mitigando tendências enviesadas e evitando sobreajuste (*overfitting*) durante o treinamento.

4.7.2 Pré-processamento dos Dados

A padronização dos atributos foi realizada utilizando a transformação `z-score`, dada por:

$$\hat{x}_i = \frac{x_i - \mu}{\sigma}$$

onde μ e σ são a média e o desvio padrão, respectivamente, calculados sobre o conjunto de treinamento $\mathcal{D}_{train}$. Essa transformação assegura que os dados possuam média zero e variância unitária, propriedade essencial para estabilizar o processo de otimização em redes neurais profundas.

Neste trabalho, a padronização foi aplicada utilizando o StandardScaler, da biblioteca Scikit-learn. Primeiramente, são calculados a média e o desvio padrão de cada atributo com base no conjunto de treinamento por meio do método `fit_transform`, que também realiza a transformação dos dados de treino. Em seguida, a mesma transformação é aplicada aos conjuntos de validação e teste por meio do método `transform`, utilizando os parâmetros previamente ajustados no treino. Essa abordagem evita vazamento de dados e assegura consistência na escala das variáveis em todos os conjuntos.

Para a etapa de avaliação, os conjuntos de validação e teste, que contêm tráfego misto (benigno e malicioso), foram transformados utilizando os mesmos parâmetros do `StandardScaler`

ajustado no treino. Em seguida, para viabilizar o cálculo das métricas de desempenho, os rótulos desses conjuntos foram convertidos em formato binário (0 para benigno, 1 para anomalia) com o `LabelEncoder`. Esse procedimento permite uma avaliação quantitativa da capacidade do modelo em generalizar e identificar anomalias em dados não vistos.

4.7.3 Arquitetura Base: HSAE (*Hybrid Scoring Autoencoder*)

Como mencionado anteriormente, a arquitetura HSAE representa a primeira proposta deste trabalho, constituindo uma arquitetura de *autoencoder* híbrida que combina aprendizado de reconstrução com classificação direta de anomalias. Esta abordagem busca superar limitações dos *autoencoders* tradicionais, que dependem exclusivamente do erro de reconstrução para detecção, incorporando uma saída auxiliar especializada em pontuação probabilística de anomalias.

A Figura 12 apresenta a metodologia adotada nesta pesquisa para a avaliação do modelo híbrido proposto. Ela descreve desde a divisão dos dados utilizados, passando pelo pré-processamento necessário, treinamento do modelo proposto (HSAE) e a etapa final de avaliação do desempenho por meio de diversas métricas.

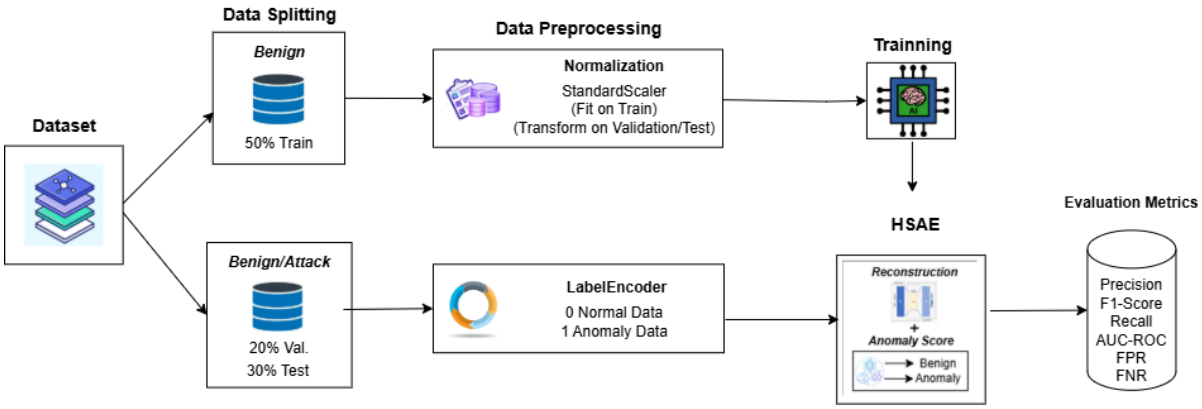


Figura 12 – Cenário de testes para o modelo proposto.

Nosso objetivo é simular um cenário realista de detecção de anomalias, onde treinamos o modelo exclusivamente com dados benignos e, posteriormente, o expomos a tráfego misto, contendo instâncias normais e maliciosas. Para isso, dividimos os dados da seguinte forma: 50% do tráfego benigno foi destinado ao treinamento, 20% foram utilizados na validação e os 30% restantes compuseram o conjunto de teste. O *threshold* é definido via EER sobre o

conjunto de validação e então aplicado ao teste.

Para os dados de treinamento, utilizamos a técnica de padronização `z-score` apresentada na seção 4.6.3 para o pré-processamento dos atributos, por meio do `StandardScaler`. Os parâmetros de média e desvio padrão foram calculados exclusivamente sobre o conjunto de treinamento, uma prática que evita o vazamento de informações (*data leakage*) entre as etapas. Esse conjunto de dados de treino, já normalizado e composto unicamente por tráfego benigno, foi então utilizado para treinar o modelo HSAE, permitindo que ele aprendesse a representar o padrão de normalidade da rede.

4.7.4 Extensão *Ensemble*: HSAE + PCA + One-Class SVM

Dando sequência à justificativa apresentada na Seção 4.2, esta seção detalha a implementação do *ensemble* híbrido, concebido para superar as limitações do modelo HSAE isolado. A arquitetura integra o *autoencoder* para aprendizado de características com um classificador de fronteira para a detecção de anomalias, buscando um *score* mais completo ao combinar a sensibilidade do erro de reconstrução com a precisão de um modelo de fronteira.

A Figura 13 ilustra a metodologia adotada nesta pesquisa. O fluxograma descreve o processo completo, desde a divisão dos dados, passando pelo pré-processamento, o treinamento do *ensemble* (HSAE+PCA e OC-SVM) e a etapa final de avaliação de desempenho por meio de um conjunto de métricas. No *pipeline* do *ensemble*, o modelo HSAE, previamente treinado com sua função de perda híbrida, é empregado de forma especializada. Embora sua saída auxiliar seja fundamental durante o treinamento para aprimorar a qualidade da representação latente, na etapa de inferência do *ensemble*, apenas duas de suas saídas são utilizadas: o erro de reconstrução, que é combinado com o *score* do One-Class SVM, e a representação latente, que serve de entrada para o *pipeline* PCA+OCSVM. Essa abordagem mantém a especialização de cada componente: o HSAE para modelagem de padrões normais e o OCSVM para definição de fronteiras de decisão.

Este componente utiliza kernel Radial Basis Function (RBF) com γ='auto' e ν=0.045, gerando *scores* de anomalia baseados na distância à fronteira.

O objetivo do desenho experimental é simular um cenário realista, onde o modelo é treinado exclusivamente com dados benignos e, posteriormente, avaliado em um ambiente com tráfego misto. Para isso, os dados foram divididos da seguinte forma: 50% do tráfego benigno foi destinado ao treinamento; 20% foram utilizados para a validação; e os 30% restantes com-

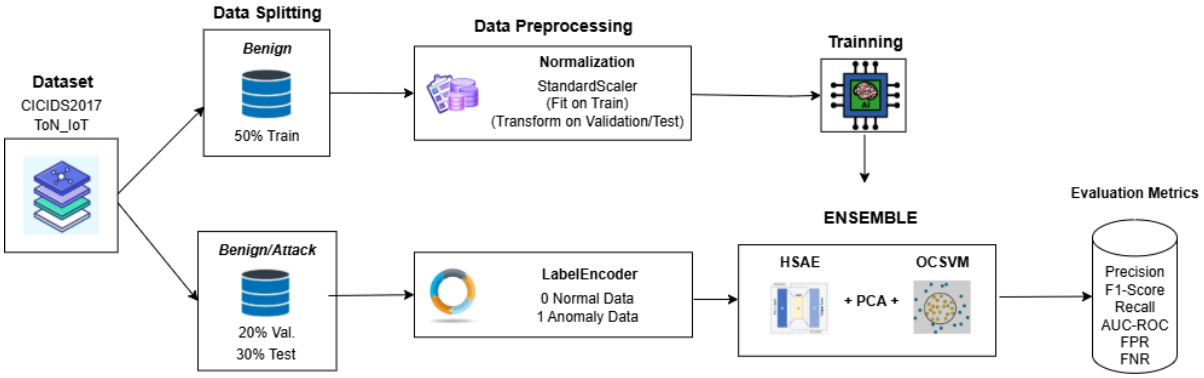


Figura 13 – Cenário de testes para o modelo ensemble.

puseram o conjunto de teste. O limiar de decisão (*threshold*) é otimizado via *Equal Error Rate* (EER) sobre o conjunto de validação e, então, aplicado ao conjunto de teste para a classificação final.

Assim como no HSAE, para os dados de treinamento, utilizou-se a técnica de padronização *z-score* apresentada na seção 4.6.3 para o pré-processamento dos atributos por meio do StandardScaler. A implementação segue etapas sequenciais: treinamento do HSAE com dados benignos, extração de *features* latentes, aplicação de PCA para redução dimensional, e treinamento do One-Class SVM. Este componente utiliza kernel RBF com γ='auto' e ν=0.045, gerando *scores* de anomalia baseados na distância à fronteira. O *score* final é uma combinação ponderada: Score_final $= 0.5 \times$ Score_HSAE $+ 0.5 \times$ Score_OneClassSVM. Ambos os *scores* são normalizados no intervalo [0,1] para garantir contribuição equilibrada, e o *threshold* é definido via EER sobre o conjunto de validação antes de ser aplicado ao teste.

Essa abordagem oferece complementariedade entre métodos: o HSAE captura padrões de reconstrução, enquanto o One-Class SVM detecta desvios latentes. A combinação mitiga fraquezas individuais e amplifica as forças de cada técnica, resultando em um sistema de detecção mais robusto e preciso.

4.7.5 Modelo de Comparação: VAE (*Variational Autoencoder*)

Seguindo a abordagem do estudo de (ZAVRAK; ISKEFIYELI, 2020). Entre os modelos avaliados por aqueles autores, incluindo o *Autoencoder* tradicional e o One-Class SVM, o *Variational Autoencoder* (VAE) apresentou os melhores resultados. Como apresentado anteriormente, o VAE continua sendo valorizado na literatura atual como uma solução eficaz para detecção

de anomalias, especialmente em contextos relacionados à segurança da informação. Conforme (BERAHMAND et al., 2024), *autoencoders* como o VAE vêm sendo empregados com frequência em tarefas que envolvem a identificação de comportamentos irregulares em dados complexos. Sua capacidade de modelar distribuições probabilísticas permite detectar desvios sutis em relação ao padrão normal, característica fundamental para sistemas de detecção de intrusões (IDS). Nesse sentido, o VAE é classificado como um *autoencoder* generativo e figura entre as abordagens mais promissoras para cenários envolvendo tráfego anômalo.

Embora o VAE seja uma referência importante no campo, o artigo de (ZAVRAK; ISKEFIYELI, 2020) não fornece descrição suficientemente detalhada do processo de preparação dos dados. Etapas essenciais, como os critérios adotados para normalização, estratégias de validação e o tratamento do desbalanceamento entre as classes, não são claramente especificadas, o que compromete a reprodutibilidade dos experimentos.

Como não dispúnhamos do código original, reimplementamos o VAE ajustado ao nosso cenário experimental, de modo a assegurar consistência metodológica em termos de pré-processamento, divisão dos dados e métricas de avaliação. Essa reimplementação preserva a estrutura central proposta por (ZAVRAK; ISKEFIYELI, 2020), com *encoder*, *decoder* e regularização do espaço latente via penalização Kullback-Leibler divergence (KL). No entanto, foram adotadas adaptações práticas, como o uso do erro de reconstrução como *score* de detecção (em substituição à probabilidade de reconstrução) (BERAHMAND et al., 2024) (YANG et al., 2022) e treinamento com o otimizador Adam (BERAHMAND et al., 2024). Tais escolhas são compatíveis com diretrizes amplamente aceitas na literatura para implementações práticas de *autoencoders* variacionais, que destacam a flexibilidade desses modelos quanto à função de ativação, tipo de perda e estratégias de limiar baseadas em métricas estatísticas do erro de reconstrução (YANG et al., 2022). Essas modificações foram necessárias para permitir uma comparação justa com o HSAE, sem comprometer os princípios fundamentais do modelo variacional.

A arquitetura implementada para o VAE inclui um encoder com camadas *Dense* (512 e 256 neurônios, *LeakyReLU*), normalização por lotes e *dropout*, espaço latente com dimensão 64, e um *decoder* simétrico. A função de perda combina MSE (*Mean Squared Error*) com penalização KL-divergence, e o treinamento foi conduzido por 150 épocas. Para a detecção de anomalias, utilizou-se o erro de reconstrução como métrica base, sendo o limiar de decisão otimizado através do EER. O EER representa o ponto operacional onde a taxa de falsos positivos (FPR) se iguala à taxa de falsos negativos (FNR), proporcionando um equilíbrio otimizado entre essas métricas (CHENG; WANG, 2004). O *threshold* EER foi calculado sobre

o conjunto de validação e posteriormente aplicado no conjunto de teste, garantindo uma avaliação mais robusta e teoricamente fundamentada em comparação com métodos baseados em percentis fixos (YANG et al., 2022). Essa abordagem permite uma comparação mais justa entre os modelos, uma vez que ambos operam sob condições de limiar otimizadas.

4.7.6 Modelo de Comparação: VAE + PCA + One-Class SVM

Para aprimorar a capacidade de detecção do modelo VAE isolado, foi desenvolvida uma extensão que implementa uma abordagem híbrida, combinando o *Variational Autoencoder* (VAE) com um classificador *One-Class Support Vector Machine* (OC-SVM). Essa arquitetura sinérgica busca superar as limitações de um sistema baseado unicamente no erro de reconstrução, criando um mecanismo de detecção de anomalias com duas camadas de análise.

A lógica fundamental deste modelo híbrido é utilizar o VAE não apenas para a reconstrução de dados, mas também como um extrator de características não-lineares. O VAE aprende a comprimir os dados de tráfego normal em um espaço latente de dimensionalidade reduzida (64 dimensões, neste caso), que captura as características mais salientes e essenciais da normalidade.

O processo de detecção ocorre em duas frentes simultâneas. Na primeira, as representações latentes geradas pelo encoder a partir dos dados de treinamento são otimizadas via Análise de Componentes Principais (PCA), que retém 95% da variância e estabiliza o processo. Em seguida, um modelo One-Class SVM é treinado com essas *features* para aprender uma fronteira de decisão que delimita a normalidade no espaço de características. Na segunda frente, o VAE reconstrói a instância de entrada, e o seu erro de reconstrução é calculado como uma segunda métrica de anomalia, que tende a ser maior para dados anômalos.

Para cada instância, o *score* de anomalia do OC-SVM e o erro de reconstrução do VAE são normalizados e combinados através de uma média ponderada. Isso gera um *score* de anomalia híbrido e final, que reflete tanto a dificuldade de reconstrução da instância quanto sua conformidade com a distribuição de dados normais no espaço latente.

4.7.7 Etapa de Treinamento

O modelo foi treinado por 150 épocas com *batch size* 128, utilizando o algoritmo de otimização Adam com taxa de aprendizado igual a 5×10^{-5}. A função de custo híbrida

garante que o modelo seja sensível a desvios sutis na estrutura do tráfego.

O treinamento segue a estratégia de aprendizado não supervisionado, utilizando exclusivamente dados normais (50% do conjunto de dados benignos, conforme descrito na seção 4.3.3). Todas as amostras benignas recebem o rótulo zero, estabelecendo o padrão de normalidade que o *autoencoder* deve aprender, sem exposição prévia a padrões maliciosos.

Durante o processo de treinamento, o modelo otimiza simultaneamente a capacidade de reconstrução e a detecção de anomalias através da função de perda híbrida definida na equação (4.3). Essa abordagem permite que o modelo aprenda representações robustas do tráfego normal, essenciais para a posterior detecção de anomalias.

4.7.8 Síntese da Proposta e Vantagens sobre as Abordagens Correlatas

Para abordar sistematicamente essas limitações, este trabalho propõe inicialmente uma arquitetura *Autoencoder* (HSAE) que incorpora uma camada adicional de classificação no espaço latente e utiliza uma função de perda híbrida, combinando reconstrução (MSE) e classificação (*binary cross-entropy*). O modelo HSAE também emprega otimização dinâmica baseada no *Equal Error Rate* (EER) para ajuste automático da fronteira de decisão, substituindo os *thresholds* fixos utilizados em trabalhos como Zavrak e Iskefiyeli (2020) por um *threshold* que se adapta automaticamente às características dos dados através de um *score* combinado que integra o erro de reconstrução com o *score* de anomalia da camada de classificação. Esta proposta busca oferecer uma representação robusta e discriminativa dos padrões normais de tráfego, estabelecendo uma *baseline* com potencial eficiência computacional e reduzindo a dependência de ajustes manuais ou processos complexos. Diferentemente de trabalhos como Zavrak et al. que se restringem à métrica AUC, o modelo é avaliado utilizando um conjunto mais amplo de métricas, incluindo precisão, *recall*, *F1-score*, FPR e FNR.

Posteriormente, o modelo é ampliado em uma abordagem *ensemble* híbrida sequencial, integrando o HSAE com PCA para redução dimensional das representações latentes e One-Class SVM para classificação das representações reduzidas. Esta arquitetura *ensemble* utiliza um *score* combinado que integra o erro de reconstrução do HSAE com o *score* de anomalia do One-Class SVM, aplicando também a otimização dinâmica baseada no *Equal Error Rate* (EER) para ajustar automaticamente a fronteira de decisão às variações no tráfego. Assim, busca-se mitigar limitações específicas observadas nos trabalhos anteriores, incluindo controle dinâmico dos falsos positivos, menor complexidade estrutural, maior cobertura na avaliação e

maior adaptabilidade.

As duas propostas trabalham em conjunto para endereçar cada limitação identificada na tabela comparativa. O modelo HSAE isolado resolve a dependência de parâmetros estáticos através do aprendizado automático de representações discriminativas e da substituição de *thresholds* fixos por otimização dinâmica do EER aplicada ao *score* combinado (erro de reconstrução + *score* de anomalia da camada de classificação), enquanto o *ensemble* híbrido sequencial (HSAE+PCA+OCSVM) amplia essa capacidade adaptativa utilizando um *score* combinado diferente (erro de reconstrução + *score* do One-Class SVM) com a mesma estratégia de *threshold* dinâmico baseado no *Equal Error Rate*, eliminando a necessidade de configuração manual de parâmetros de detecção. A arquitetura simplificada contrasta com a complexidade excessiva de modelos como F-OSFA, e o processo automatizado de combinação de *scores* com otimização EER elimina a necessidade de ajustes manuais presentes em trabalhos como Zahoora et al. A avaliação abrangente com múltiplas métricas supera a limitação de trabalhos que utilizam apenas AUC-ROC.

As propostas deste trabalho buscam contribuir com um avanço incremental na área, fundamentado diretamente nas lacunas observadas na literatura e sistematizadas na tabela comparativa apresentada. Espera-se oferecer maior adaptabilidade através de mecanismos automáticos de ajuste baseados em EER e *scores* combinados, menor complexidade estrutural através da arquitetura híbrida sequencial HSAE+PCA+OCSVM mantendo eficácia na detecção, avaliação mais abrangente com múltiplas métricas de desempenho, e aplicabilidade prática em cenários dinâmicos e críticos como redes IoT e infraestruturas industriais.

Conforme discutido no Capítulo 3, a análise dos trabalhos relacionados revelou um conjunto de limitações que motivaram esta pesquisa. As arquiteturas do HSAE e de sua extensão *ensemble* foram concebidas para superar diretamente cada uma dessas deficiências. A Tabela 1, a seguir, sistematiza essa relação, ilustrando como cada limitação observada é abordada por uma característica específica dos modelos aqui propostos.

Tabela 1 – Limitações Identificadas versus Soluções Propostas

Limitação Identificada	**Trabalhos Afetados**	**Solução Proposta Nesta Pesquisa**
Falta de mecanismos adaptativos	(ZAVRAK; ISKEFIYELI, 2020), (MBONA; ELOFF, 2022))	**Threshold dinâmico via EER** em substituição aos *thresholds* fixos, permitindo ajuste automático da fronteira de decisão baseado nas características dos dados, combinado com *scores* adaptativos que integram múltiplas fontes de informação
Dependência de parâmetros estáticos	(MBONA; ELOFF, 2022), (SOLTANI et al., 2023)	**Função de perda híbrida** com aprendizado automático de representações discriminativas e *scores* combinados adaptativos
Complexidade arquitetural excessiva	(LU et al., 2024)), (MINHAS et al., 2025), (SOLTANI et al., 2023)	**Arquitetura híbrida sequencial** HSAE+PCA+OCSVM com redução dimensional
Necessidade de ajustes manuais	(ZAHOORA et al., 2022), (SOLTANI et al., 2023)	**Processo automatizado** de combinação de *scores* (reconstrução + classificação/OCSVM) com otimização EER
Avaliação limitada	(ZAVRAK; ISKEFIYELI, 2020)	**Conjunto amplo de métricas** incluindo precisão, *recall, F1-score*, FPR e FNR

5 RESULTADOS COMPARATIVOS

Este capítulo dedica-se à apresentação e análise dos resultados experimentais obtidos para validar as propostas desta dissertação. O desempenho do modelo HSAE e de sua extensão *Ensemble* é rigorosamente avaliado por meio de uma análise comparativa com suas respectivas contrapartes baseadas no modelo de referência VAE. A validação ocorre em dois cenários distintos, representados pelos conjuntos de dados CICIDS2017 e ToN_IoT. Por fim, a eficiência computacional das arquiteturas é investigada para aferir sua viabilidade em ambientes com recursos limitados.

5.1 COMPARAÇÃO DE DESEMPENHO ENTRE OS MODELOS HSAE E VAE PARA O CONJUNTO DE DADOS CICIDS2017

Nesta seção obtivemos o desempenho do HSAE, utilizando o mesmo conjunto de dados CICIDS2017 empregado no estudo sobre o VAE de (ZAVRAK; ISKEFIYELI, 2020). Para garantir uma comparação justa e atualizada, realizamos uma reimplementação metodológica do modelo VAE, detalhada na seção 4.6.6, incorporando técnicas recentemente validadas pela literatura científica atual. Essa atualização metodológica resultou em melhorias no desempenho geral do modelo adaptado para detecção dos ataques avaliados. Ao comparar os valores da métrica AUC obtidos por nosso modelo VAE atualizado com aqueles gerados pela implementação original de Zavrak e Iskefiyeli (2020), observou-se um ganho em praticamente todos os ataques considerados. Embora para o ataque DoS Slowloris não tenha havido alteração significativa no valor da AUC (0.87), houve aumento nos ataques DoS GoldenEye (de 0.80 para 0.92), DoS Hulk (de 0.81 para 0.91) e DoS SlowHTTPTest (de 0.86 para 0.94). Essas melhorias demonstram a eficácia das técnicas introduzidas, destacando-se o uso do erro de reconstrução como métrica de detecção, a otimização do *threshold* baseada no EER, bem como as melhorias arquiteturais na rede neural empregada. Além disso, esses resultados reforçam a importância da adoção de abordagens metodológicas modernas na área de segurança cibernética, permitindo avaliações mais robustas e eficazes em cenários reais.

Além da métrica AUC adotada por Zavrak e Iskefiyeli (ZAVRAK; ISKEFIYELI, 2020), ampliamos a análise comparativa entre o VAE e o modelo HSAE, utilizando métricas adicionais relevantes, tais como precisão, *recall*, *F1-score*, taxa de falsos positivos (FPR) e taxa de falsos

negativos (FNR). A Tabela 2 apresenta os resultados obtidos para cada tipo de ataque de negação de serviço (DoS e DDoS), assim como para cenários de múltiplos ataques simultâneos, considerando o conjunto de dados CICIDS2017, com os melhores valores de cada métrica destacados em negrito para cada um dos dois modelos comparados. Essa abordagem visa garantir uma comparação justa e metodologicamente alinhada com os objetivos da proposta.

Tabela 2 – Comparação de desempenho para o conjunto de dados CICIDS2017.

Ataque	Modelo	Precisão	FPR	FNR	Recall	F1-Score	AUC
DDoS	VAE	65%	34,76%	**34,49%**	**66%**	65%	0.75
	HSAE	**91%**	**6,36%**	35,95%	64%	**75%**	**0.82**
DoS GoldenEye	VAE	87%	13,02%	**13,93%**	**86%**	**86%**	0.92
	HSAE	**90%**	**8,82%**	22,61%	77%	83%	0.92
DoS Hulk	VAE	80%	20,10%	**20,32%**	**80%**	80%	0.91
	HSAE	**90%**	**8,30%**	26,53%	73%	**81%**	**0.94**
DoS SlowHTTPTest	VAE	91%	8,95%	**11,71%**	**88%**	**90%**	0.94
	HSAE	91%	**8,73%**	13,88%	86%	88%	0.94
DoS Slowloris	VAE	76%	**24,21%**	21,79%	78%	77%	0.87
	HSAE	**77%**	27,40%	**10,28%**	**90%**	**83%**	**0.89**
Múltiplos Ataques	VAE	78%	23,42%	18,03%	82%	80%	0.88
	HSAE	**80%**	**22,27%**	**8,22%**	**92%**	**86%**	**0.93**

5.1.1 Análise Comparativa com ataques isolados

Os resultados revelam que o modelo HSAE apresenta desempenho superior em diversas métricas relevantes. Por exemplo, no caso do ataque DDoS, o HSAE alcançou 91% de precisão, superando os 65% do VAE, com uma melhoria significativa no *F1-score* (75% contra 65%). Apesar do *recall* ter tido uma leve inferioridade (64% vs 66%), o HSAE apresentou uma boa redução na FPR (6,36% contra 34,76%), demonstrando excelente controle de falsos positivos. A métrica AUC do HSAE apresentou valor de 0,82, superando o VAE que obteve 0,75, refletindo uma maior capacidade do HSAE em classificar corretamente o tráfego malicioso e benigno.

No ataque DoS GoldenEye, ambos os modelos apresentaram desempenho equilibrado, com o HSAE obtendo 90% de precisão contra 87% do VAE. O *F1-score* do HSAE foi ligeiramente inferior (83% vs 86%), assim como o *recall* (77% vs 86%). A FPR do HSAE foi de 8,82%, comparada aos 13,02% do VAE, evidenciando melhor controle de falsos positivos. A métrica AUC manteve-se equivalente em 0,92 para ambos os modelos, demonstrando capacidade similar de

discriminação entre as classes neste tipo específico de ataque.

Em relação ao ataque DoS Hulk, o HSAE demonstrou melhorias expressivas, com *F1-score* de 81% superando os 80% do VAE, precisão de 90% contra 80%. O *recall* apresentou leve redução (73% vs 80%), porém a FPR do HSAE foi bastante inferior (8,30% vs 20,10%), demonstrando melhor controle de falsos positivos. A métrica AUC do HSAE foi superior (0,94 contra 0,91 do VAE), indicando melhor desempenho geral na discriminação entre tráfego benigno e malicioso para este tipo de ataque.

Nos ataques DoS Slowhttptest, o HSAE apresentou resultados superiores com 91% de precisão, mesmo valor do VAE, mas com *F1-score* inferior (88% vs 90%). O *recall* do HSAE foi de 86%, ligeiramente inferior aos 88% do VAE, enquanto a FPR ficou em 8,73% contra 8,95% do VAE. A métrica AUC manteve-se equivalente em 0,94 para ambos os modelos, demonstrando capacidade similar de classificação.

No caso do DoS Slowloris, os resultados apresentaram maior variabilidade. O HSAE obteve 77% de precisão, superior aos 76% do VAE, com *F1-score* superior (83% vs 77%) e *recall* expressi- vamente melhor (90% vs 78%). A FPR do HSAE foi de 27,40%, superior aos 24,21% do VAE, indicando maior taxa de falsos positivos neste cenário específico. A métrica AUC do HSAE foi de 0,89, superior aos 0,87 do VAE.

Para avaliar a robustez dos modelos em cenários mais complexos e realistas, foi conduzida uma análise adicional envolvendo múltiplos tipos de ataques simultâneos. Nesta abordagem, foi utilizada uma metodologia de divisão de dados onde o conjunto de validação incluiu ataques DDoS, DoS Slowloris e DoS Slowhttptest, enquanto o conjunto de teste foi composto por DDoS, DoS Slowloris, DoS Slowhttptest, DoS Hulk e DoS GoldenEye. Esta metodologia permite avaliar tanto a capacidade dos modelos de detectar ataques individuais quanto sua performance em ambientes com múltiplas ameaças coexistentes e ataques também não vistos na validação.

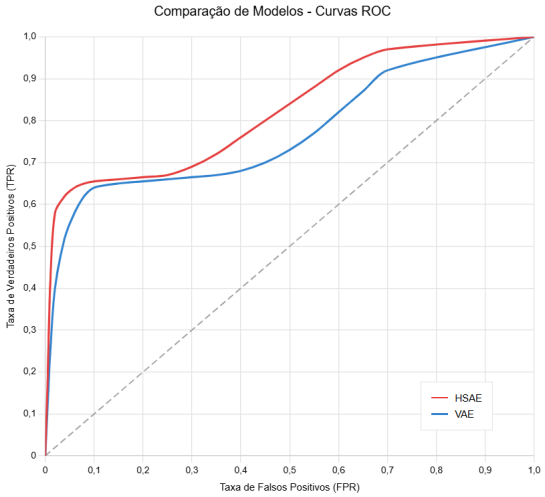
Os resultados dos múltiplos ataques revelam aspectos importantes sobre a generalização dos modelos. O HSAE demonstrou 80% de precisão contra 78% do VAE, com *F1-score* superior (86% vs 80%). O *recall* do HSAE foi notavelmente superior (92% vs 82%), evidenciando maior capacidade de identificação de ataques em cenários complexos. Embora a FPR tenha superado o VAE (22,27% vs 23,42%), a diferença é pequena. A métrica AUC do HSAE foi melhor (0,93 contra 0,88 do VAE), apresentando uma capacidade superior de discriminação em ambientes com múltiplas ameaças.

5.1.2 Visualização Comparativa dos Resultados

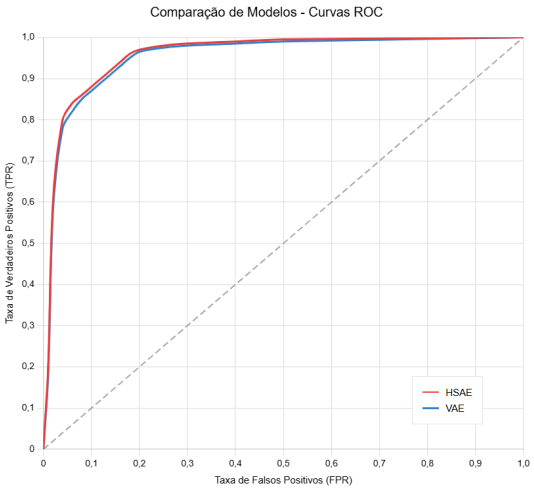
As curvas ROC apresentadas na Figura 14 ilustram a separação entre tráfego benigno e malicioso nos diferentes ataques, evidenciando visualmente o desempenho dos modelos analisados. Essa distinção permite uma comparação direta entre as abordagens, destacando o desempenho superior do HSAE em termos de capacidade de discriminação, especialmente em cenários com ataques variados.

Os resultados indicam que o HSAE mantém um desempenho eficaz mesmo em cenários complexos, nos quais diferentes tipos de ataques ocorrem simultaneamente. A comparação entre ataques isolados e múltiplos ataques revela que, apesar de variações pontuais em algumas métricas, o modelo preserva sua robustez, especialmente nos valores de *recall* e AUC, métricas necessárias para a detecção de anomalias em ambientes reais.

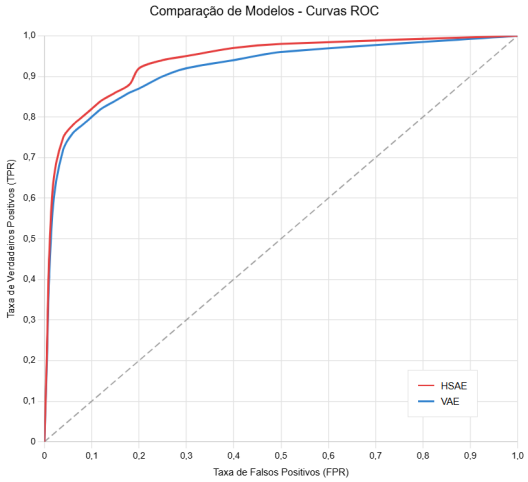
Por fim, com o *dataset* CICIDS2017, o modelo HSAE demonstrou-se mais eficiente em termos gerais, validando sua adoção frente a abordagens tradicionais baseadas exclusivamente em reconstrução. Os resultados obtidos, particularmente em cenários de múltiplos ataques, reforçam o potencial do HSAE para aplicações reais em ambientes corporativos, onde a detecção confiável de diversas ameaças simultâneas é fator fundamental.



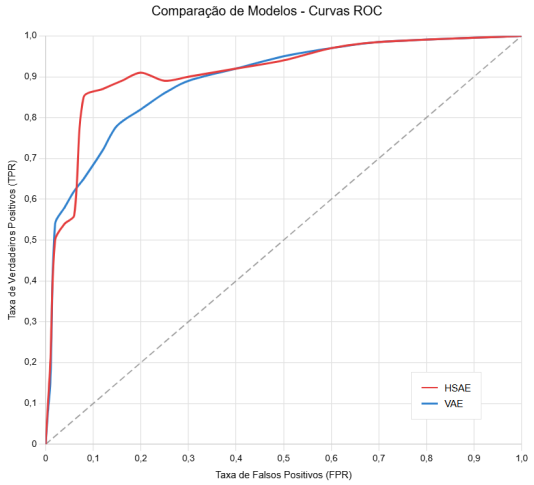
(a) Curva ROC para DDoS



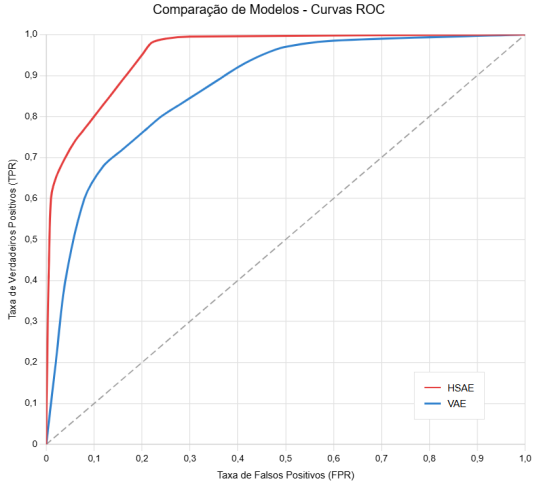
(b) Curva ROC para DoS Slowhttptest



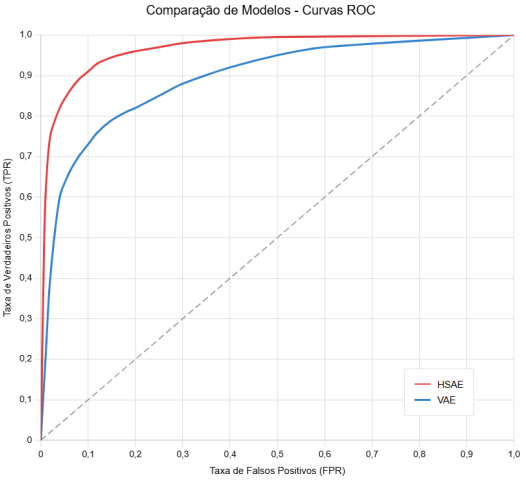
(c) Curva ROC para DoS GoldenEye



(d) Curva ROC para DoS Slowloris



(e) Curva ROC para DoS Hulk



(f) Curva ROC para Múltiplos Ataques

Figura 14 – Comparação das curvas ROC para os modelos HSAE e o VAE usando o *dataset* CICIDS2017.

5.2 COMPARAÇÃO DE DESEMPENHO ENTRE OS MODELOS HSAE E VAE PARA O CONJUNTO DE DADOS TON_IOT

Foram realizados testes utilizando o *ToN_IoT*, um conjunto de dados que reflete padrões de tráfego oriundos de dispositivos IoT em ambientes industriais e residenciais, abrangendo uma variedade de vetores de ataque e condições realistas de operação. A Tabela 3 apresenta os resultados obtidos para três classes de ataques distintas, comparando o desempenho do modelo proposto HSAE com o modelo de referência.

Tabela 3 – Comparação de desempenho para o conjunto de dados ToN_IoT.

Ataque	**Modelo**	**Precisão**	**FPR**	**FNR**	***Recall***	***F1-Score***	**AUC**
DDoS	VAE	70%	25,69%	25,46%	*75%*	*72%*	0.79
	HSAE	**85%**	**14,77%**	**15,55%**	***84%***	***85%***	**0.91**
DoS	VAE	69%	26,34%	26,78%	*73%*	*71%*	0.76
	HSAE	**91%**	**22,11%**	**22,13%**	***78%***	***84%***	**0.77**
Ransomware	VAE	67%	28,09%	27,98%	*71%*	*70%*	0.76
	HSAE	**86%**	**13,93%**	**15,07%**	***85%***	***85%***	**0.86**
Múltiplos Ataques	VAE	85%	20,30%	43,69%	*56%*	*68%*	0.81
	HSAE	**95%**	**10,31%**	**9,71%**	***90%***	***92%***	**0.89**

5.2.1 Análise Comparativa com ataques individuais

No contexto do ataque DDoS, o modelo HSAE demonstrou ganhos consideráveis em relação ao VAE. A precisão aumentou de 70% para 85%, evidenciando que o processo de codificação-decodificação variacional apresenta limitações na captura completa de padrões característicos do *dataset* ToN_IoT. A taxa de falsos positivos (FPR) apresentou redução expressiva de 25,69% para 14,77%, demonstrando que ambos os modelos alcançam patamares operacionalmente viáveis, porém o HSAE mantém vantagem significativa em ambientes com baixa tolerância a alarmes indevidos. A taxa de falsos negativos (FNR) reduziu de 25,46% para 15,55%, indicando que o VAE apresenta maior propensão à não identificação de ataques legítimos comparado ao HSAE. O *F1-score* aumentou de 72% para 85%, e o *recall* evoluiu de 75% para 84%, evidenciando que a arquitetura híbrida proporciona cobertura superior na identificação dos ataques. O valor da AUC passou de 0,79 para 0,91, representando um salto qualitativo expressivo na capacidade discriminativa, sugerindo que a arquitetura híbrida do

HSAE é particularmente efetiva na separação entre tráfego benigno e malicioso.

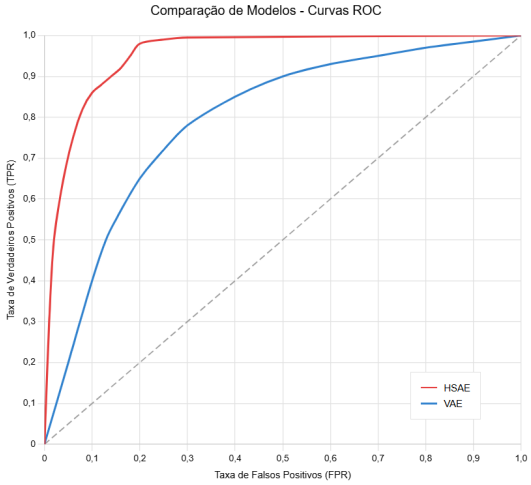
Para ataques DoS, observou-se comportamento similar com nuances interessantes, onde a precisão elevou-se de 69% para 91%, demonstrando superioridade consistente do HSAE. O FPR diminuiu de 26,34% para 22,11%, indicando que ambos os modelos operam em faixas de falsos positivos relativamente controladas, com o HSAE apresentando vantagem operacional. O *recall* apresentou aumento de 73% para 78%, evidenciando que o HSAE proporciona cobertura superior na detecção de ataques. O *F1-score* progrediu de 71% para 84%, e a AUC demonstrou evolução de 0,76 para 0,77, confirmando o aprimoramento da capacidade de distinção entre tráfego legítimo e malicioso, embora a diferença seja menor neste cenário específico, possivelmente devido às características menos complexas dos ataques DoS em comparação com DDoS.

No cenário de ataques *Ransomware*, o HSAE manteve a tendência de superioridade, com diferenças proporcionalmente menores, evidenciando que o VAE demonstra competência considerável neste domínio específico. A precisão aumentou de 67% para 86%, e o FPR reduziu de 28,09% para 13,93%, isso indica que ambos os modelos alcançam níveis de desempenho relevantes para aplicações práticas, com o HSAE se destacando por apresentar resultados superiores. O *recall* evoluiu de 71% para 85%, acompanhado pelo *F1-score* que progrediu de 70% para 85%. A AUC apresentou melhoria de 0,76 para 0,86, sugerindo que as características específicas dos ataques *ransomware* são adequadamente capturadas por ambas as arquiteturas, com vantagem maior para o modelo híbrido.

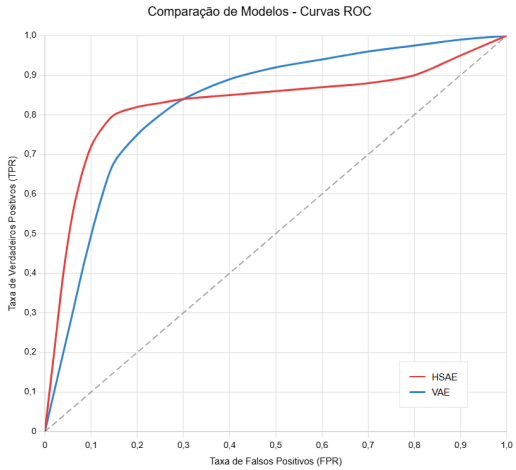
Na análise de múltiplos ataques, onde a validação foi conduzida com ataques DoS, e o teste realizado com ataques DDoS e *ransomware*, observou-se a menor diferença relativa entre os modelos, indicando robustez considerável do VAE em cenários de generalização. O modelo HSAE demonstrou desempenho superior, com precisão alcançando 95% comparada aos 85% do VAE, mantendo diferença de aproximadamente 10 pontos percentuais. O FPR foi reduzido de 20,30% para 10,31%, demonstrando que ambos os modelos operam em faixas aceitáveis, com o HSAE proporcionando maior confiabilidade operacional. Destaca-se a redução do FNR de 43,69% para 9,71%, evidenciando que o VAE apresenta limitações na detecção de ataques verdadeiros em cenários generalizados. O *recall* atingiu 90% versus 56% do VAE, representando a maior discrepância observada e sugerindo que a capacidade de generalização é o principal diferencial arquitetural do HSAE. O *F1-score* alcançou 92% comparado aos 68% do VAE, e a AUC evoluiu de 0,81 para 0,89, confirmando a robustez superior do modelo híbrido em cenários diversificados.

5.2.2 Visualização Comparativa dos Resultados

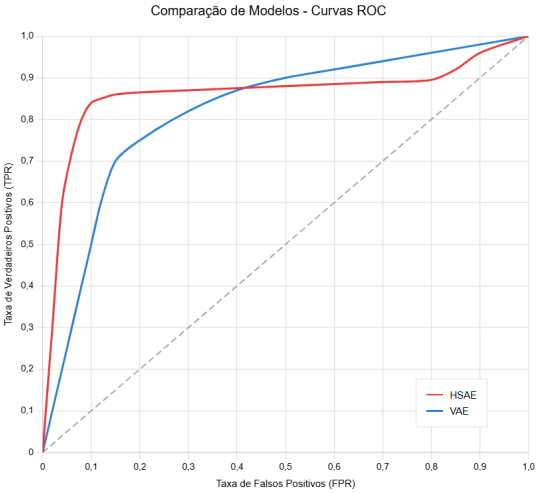
As curvas ROC apresentadas na Figura 15 ilustram a separação entre tráfego benigno e malicioso nos diferentes ataques, evidenciando visualmente o desempenho dos modelos analisados. A linha vermelha representa o modelo 1, correspondente ao HSAE (*Hybrid Scoring Autoencoder*). Já a linha azul representa o modelo 2, correspondente ao VAE (*Variational Autoencoder*).



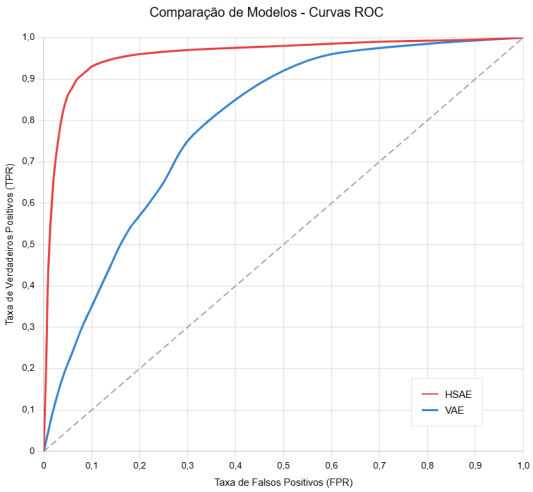
(a) Curva ROC para DDoS



(b) Curva ROC para DoS



(c) Curva ROC para Ramsoware



(d) Curva ROC para Múltiplos Ataques

Figura 15 – Comparação das curvas ROC para os modelos HSAE e VAE usando o *dataset* ToN_IoT.

Os melhores resultados do HSAE em todos os tipos de ataques avaliados sugerem que as limitações observadas do VAE estão relacionadas às características próprias do *dataset* ToN_IoT, que trabalha com dados de tráfego de rede IoT com complexidade temporal e espacial específica, onde a arquitetura híbrida do HSAE demonstra maior adequação para este

tipo de dados.

5.3 COMPARAÇÃO DE DESEMPENHO ENTRE OS *ENSEMBLES* NO CONJUNTO DE DADOS CICIDS2017.

A avaliação da arquitetura *ensemble* HSAE+PCA+One-Class SVM representa uma extensão natural dos experimentos anteriores, visando investigar se a combinação de múltiplas técnicas não supervisionadas pode superar as limitações identificadas no modelo isolado HSAE. Esta abordagem *ensemble* busca aproveitar a complementariedade entre diferentes paradigmas de detecção: o aprendizado de reconstrução do HSAE, a redução dimensional do PCA, e a modelagem de fronteiras de normalidade do One-Class SVM.

A Tabela 4 apresenta os resultados obtidos comparando duas arquiteturas *ensemble: Ensemble* HSAE (HSAE+PCA+One-Class SVM) versus *Ensemble* VAE (VAE+PCA+One-Class SVM), ambos utilizando a mesma metodologia experimental e configuração *ensemble* aplicada nas seções anteriores. Esta análise permitirá identificar os ganhos efetivos proporcionados pela substituição do VAE pelo HSAE como componente base na arquitetura *ensemble*, além de avaliar os *trade-offs* entre complexidade computacional e desempenho quando comparado ao modelo HSAE isolado da Tabela 2.

Tabela 4 – Comparação de desempenho dos *Ensembles* para o conjunto de dados CICIDS2017.

Ataque	Modelo	Precisão	FPR	FNR	Recall	F1-Score	AUC
DDoS	Ens.VAE	71%	30,72%	25,20%	75%	73%	0.81
	Ens.HSAE	**95%**	**4,53%**	**19,03%**	**81%**	**87%**	**0.87**
DoS GoldenEye	Ens.VAE	79%	21,31%	21,37%	79%	79%	0.86
	Ens.HSAE	**92%**	**8,52%**	**7,58%**	**92%**	**92%**	**0.93**
DoS Hulk	Ens.VAE	71%	31,28%	24,12%	76%	73%	0.81
	Ens.HSAE	**77%**	**28,37%**	**2,70%**	**97%**	**86%**	**0.95**
DoS SlowHTTPTest	Ens.VAE	89%	11,03%	13,03%	87%	88%	0.93
	Ens.HSAE	**97%**	**2,73%**	**3,55%**	**96%**	**97%**	**0.97**
DoS Slowloris	Ens.VAE	84%	11,79%	38,76%	61%	71%	0.78
	Ens.HSAE	**91%**	**9,55%**	**3,11%**	**97%**	**94%**	**0.94**
Múltiplos Ataques	Ens.VAE	87%	24,12%	31,17%	69%	77%	0.79
	Ens.HSAE	**96%**	**19,00%**	**8,21%**	**92%**	**94%**	**0.94**

5.3.1 Análise Comparativa Com ataques individuais

Ataques DDoS: O *Ensemble* HSAE demonstrou superioridade expressiva sobre o *Ensemble* VAE, alcançando 95% de precisão comparado aos 71% do *ensemble* VAE, representando um ganho de 24 pontos percentuais. A taxa de falsos positivos foi drasticamente reduzida de 30,72% (*Ensemble* VAE) para 4,53% (*Ensemble* HSAE), evidenciando controle superior de alarmes indevidos - aspecto crítico para a viabilidade operacional. O *recall* aumentou de 75% para 81%, enquanto o *F1-score* evoluiu de 73% para 87%. A métrica AUC progrediu de 0,81 para 0,87, confirmando maior capacidade discriminativa. Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 2: 90% precisão, 0,82 AUC), o *ensemble* apresenta ganhos incrementais mas consistentes: $+5$ pontos percentuais em precisão e melhoria substancial no controle de falsos positivos (6,36% no isolado vs 4,53% no *ensemble*).

DoS GoldenEye: Neste cenário, ambos os *ensembles* demonstraram desempenho robusto, com o *Ensemble* HSAE alcançando 92% de precisão contra 79% do *Ensemble* VAE. A FPR foi reduzida de 21,31% para 8,52%, mantendo excelente controle de falsos positivos. O *recall* permaneceu elevado em 92%, superando os 79% do *ensemble* VAE, enquanto o *F1-score* atingiu 92% comparado aos 79% do *baseline ensemble*. A AUC evoluiu de 0,86 para 0,93, indicando capacidade discriminativa superior.

Em relação ao HSAE isolado (Tabela 2: 89% precisão, 0,92 AUC), o *Ensemble* HSAE apresenta melhorias sutis, mas consistentes: $+3$ pontos percentuais em precisão e $+0{,}01$ na AUC, evidenciando robustez incremental.

DoS Hulk: O *Ensemble* HSAE demonstrou 77% de precisão, superando os 71% do *Ensemble* VAE, com FPR controlada em 28,37% versus 31,28% do *ensemble* VAE. O destaque está no *recall* excepcional de 97% comparado aos 76% do *ensemble* VAE, resultando em *F1-score* superior (86% vs 73%). A AUC atingiu 0,95, superando significativamente os 0,81 do *ensemble* VAE.

Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 2: 89% precisão, 73% recall, 0,94 AUC), observa-se um *trade-off* estratégico interessante: redução na precisão (89% $\rightarrow$ 77%, -12 pontos percentuais) em contrapartida a um ganho substancial no *recall* (73% $\rightarrow$ 97%, $+24$ pontos percentuais). Este comportamento sugere que o *ensemble* está adotando uma postura mais conservadora na detecção, priorizando a captura de ataques verdadeiros, mesmo ao custo de gerar mais falsos positivos.

Esta estratégia é particularmente relevante para ataques DoS Hulk, que são caracterizados

por seu volume elevado e potencial destrutivo significativo. Neste contexto, o custo de não detectar um ataque real (falso negativo) supera o custo operacional de investigar alarmes adicionais (falsos positivos). O aumento no *F1-score* (80% $\rightarrow$ 86%) e na AUC (0,94 $\rightarrow$ 0,95) confirma que, apesar do *trade-off* precision/recall, o *ensemble* mantém desempenho geral superior, indicando que a arquitetura consegue encontrar um ponto operacional mais adequado para este tipo específico de ataque volumétrico.

DoS SlowHTTPTest: Representa o melhor desempenho absoluto do *Ensemble* HSAE, com 97% de precisão superando os 89% do *Ensemble* VAE. A FPR foi reduzida para 2,73%, demonstrando controle excepcional de falsos positivos. O *recall* alcançou 96%, superior aos 87% do *ensemble* VAE, resultando em *F1-score* de 97% contra 88%. A AUC atingiu 0,97, próxima ao valor ideal. Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 2: 90% precisão, 0,94 AUC), o *ensemble* apresenta ganhos substanciais em todas as métricas: $+7$ pontos percentuais em precisão e $+0{,}03$ na AUC, evidenciando particular eficácia da arquitetura *ensemble* para este tipo de ataque.

DoS Slowloris: O *Ensemble* HSAE alcançou 91% de precisão, superando significativamente os 84% do *Ensemble* VAE. A FPR foi controlada em 9,55% versus 11,79% do *ensemble* VAE, enquanto o *recall* atingiu impressionantes 97% comparado aos 61% do *ensemble* VAE. O *F1-score* evoluiu para 94% contra 71%, e a AUC atingiu 0,94 versus 0,78.

Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 2: 76% precisão, 0,89 AUC), o *ensemble* apresenta os maiores ganhos observados: $+15$ pontos percentuais em precisão e $+8$ pontos em *recall*, demonstrando particular adequação da arquitetura *ensemble* para ataques mais difíceis de ser detectados.

5.3.2 Múltiplos Ataques: Robustez em Cenários Complexos

A avaliação com múltiplos ataques simultâneos revelou aspectos fundamentais sobre a robustez dos *ensembles*. O *Ensemble* HSAE demonstrou 96% de precisão contra 87% do *Ensemble* VAE, evidenciando capacidade superior de manter performance em ambientes complexos. A FPR foi reduzida de 24,12% para 19,00%, enquanto a FNR apresentou melhoria dramática de 31,17% para 8,21%, indicando detecção superior de ataques verdadeiros. O *recall* atingiu 92% versus 69% do *ensemble* VAE, resultando em *F1-score* de 94% contra 77%. A AUC evoluiu de 0,79 para 0,94, representando um salto qualitativo na capacidade discriminativa.

Comparando o *Ensemble* HSAE com o HSAE isolado em cenários de múltiplos ataques

(Tabela 2: 80% precisão, 0,93 AUC), o *ensemble* apresenta ganhos expressivos: +16 pontos percentuais em precisão e +1 ponto em *recall*. Esta melhoria sugere que a arquitetura *ensemble* é particularmente eficaz em cenários heterogêneos, onde diferentes tipos de ataques coexistem.

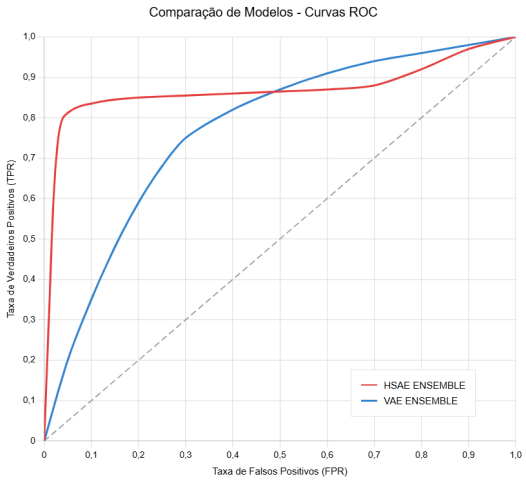
5.3.3 Visualização Comparativa dos Resultados

Para complementar a análise quantitativa apresentada, a Figura 16 apresenta as curvas ROC comparativas entre as abordagens avaliadas: *Ensemble* VAE e *Ensemble* HSAE. As curvas ROC permitem uma visualização clara da capacidade discriminativa de cada modelo, evidenciando tanto as melhorias na separação entre tráfego benigno e malicioso proporcionadas pela arquitetura *ensemble* quanto a superioridade consistente sobre o *ensemble* VAE. As curvas ilustram graficamente a relação entre True Positive Rate (TPR) e Taxa de False Positive Rate (FPR) para cada tipo de ataque, permitindo identificar visualmente os cenários onde o *ensemble* HSAE demonstra maior vantagem na capacidade de classificação. Esta representação visual facilita a compreensão da eficácia discriminativa das diferentes abordagens e reforça as conclusões da análise de AUC-ROC apresentada anteriormente, onde valores mais próximos ao canto superior esquerdo indicam desempenho superior.

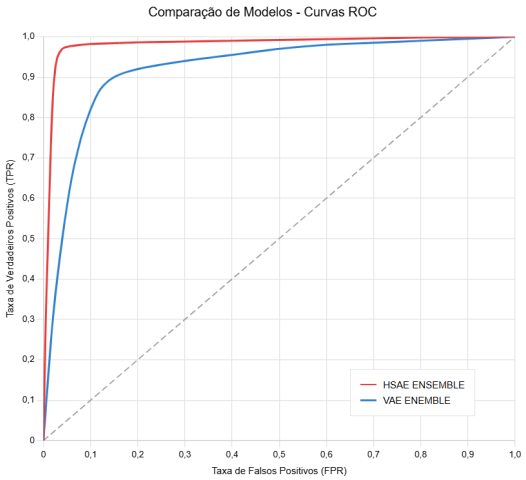
Os resultados evidenciam que a arquitetura *Ensemble* HSAE (HSAE+PCA+One-Class SVM) consegue capturar aspectos complementares dos padrões anômalos que escapam tanto ao *Ensemble* VAE quanto ao HSAE isolado. O componente PCA demonstra eficácia na redução de ruído das representações latentes extraídas pelo HSAE, preservando informações discriminativas relevantes. O One-Class SVM atua como um classificador de fronteira que identifica *outliers* no espaço reduzido, proporcionando uma segunda linha de detecção baseada em princípios geométricos distintos do aprendizado de reconstrução.

A superioridade do *Ensemble* HSAE sobre o *Ensemble* VAE confirma que o modelo base HSAE fornece representações latentes mais adequadas para a detecção de anomalias quando combinado com técnicas complementares. A combinação ponderada 50/50 entre os *scores* do HSAE e do One-Class SVM mostrou-se equilibrada, evitando dominância excessiva de qualquer componente. Esta estratégia permite que o *ensemble* capture tanto anomalias baseadas em erro de reconstrução quanto desvios geométricos da região de normalidade, resultando em maior robustez frente à diversidade de padrões anômalos.

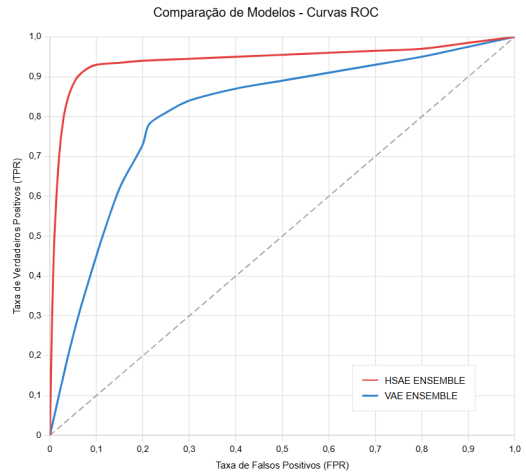
A arquitetura *ensemble* apresenta *trade-offs* importantes que devem ser considerados em aplicações práticas. O ganho em precisão e robustez vem acompanhado de maior complexidade



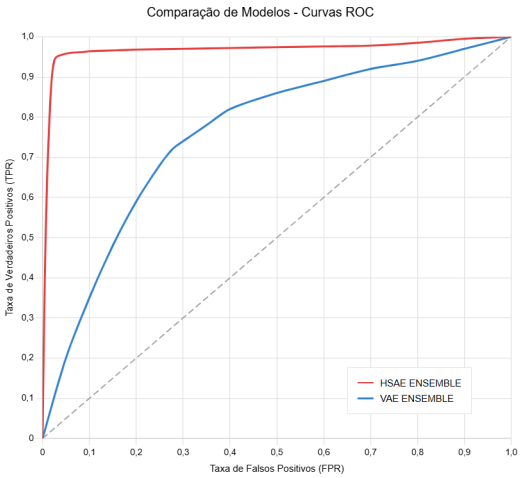
(a) Curva ROC para DDoS



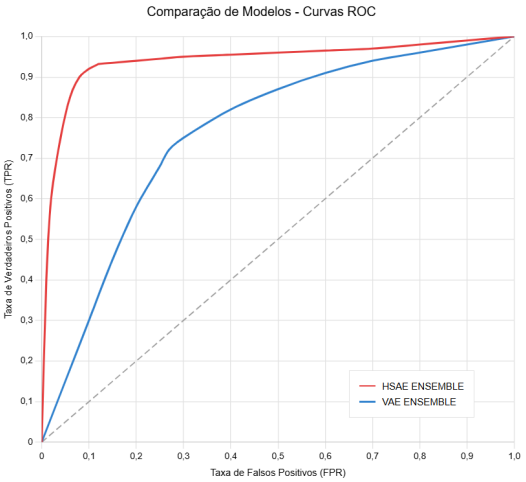
(b) Curva ROC para DoS Slowhttptest



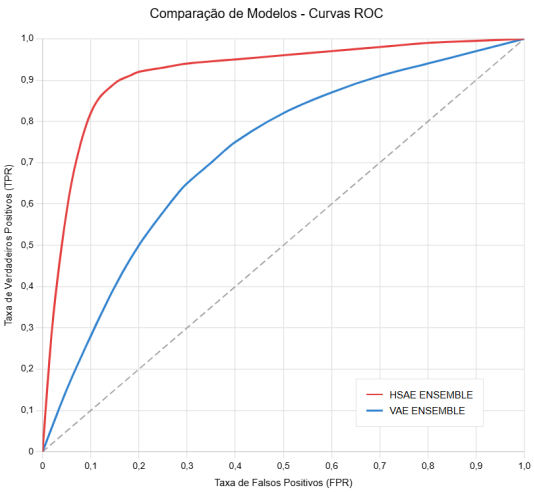
(c) Curva ROC para DoS GoldenEye



(d) Curva ROC para DoS Slowloris



(e) Curva ROC para DoS Hulk



(f) Curva ROC para Múltiplos Ataques

Figura 16 – Comparação das curvas ROC para o modelos *Ensemble* HSAE e *Ensemble* VAE usando o *dataset* CICIDS2017.

computacional, envolvendo três estágios sequenciais: extração de *features* pelo HSAE, redução dimensional via PCA, e classificação pelo One-Class SVM. Esta complexidade adicional resulta em maior latência de detecção e consumo de recursos computacionais.

Para ataques como DoS SlowHTTPTest e Slowloris, onde o *ensemble* demonstra ganhos substanciais (7-15 pontos percentuais em precisão), a complexidade adicional se justifica pela melhoria significativa na qualidade da detecção. Entretanto, para ataques como DoS Hulk, onde o *trade-off* precision/recall é menos favorável, a escolha entre modelo isolado e *ensemble* deve considerar os requisitos específicos da aplicação.

5.3.4 Síntese Estratégica

Os resultados do *Ensemble* HSAE (HSAE+PCA+One-Class SVM) estabelecem uma alternativa robusta que supera tanto o *Ensemble* VAE quanto oferece melhorias significativas sobre o HSAE isolado em cenários que demandam alta precisão e baixas taxas de falsos positivos. A arquitetura demonstra particular eficácia em:

1. **Ataques de baixa intensidade** (Slowloris): onde o *ensemble* HSAE apresenta ganhos de $+15\%$ em precisão sobre o modelo isolado e $+7\%$ sobre o *ensemble* VAE

2. **Cenários de múltiplos ataques**: onde a diversidade de padrões maliciosos exige abordagens mais sofisticadas, com o *ensemble* HSAE superando o *ensemble* VAE em 9 pontos percentuais de precisão

3. **Ambientes críticos**: onde falsos positivos representam custos operacionais elevados, com redução consistente da FPR em todos os tipos de ataque

A escolha entre HSAE isolado, *Ensemble* HSAE e *Ensemble* VAE deve ser orientada pelos requisitos específicos do ambiente de implantação, balanceando precisão desejada, recursos computacionais disponíveis e tolerância à latência de detecção. Os resultados sugerem que o *Ensemble* HSAE representa uma evolução natural tanto do HSAE isolado quanto uma alternativa superior ao *ensemble* VAE para aplicações que priorizem máxima precisão e robustez em detrimento da simplicidade computacional.

5.4 COMPARAÇÃO DE DESEMPENHO ENTRE OS *ENSEMBLES* NO CONJUNTO DE DADOS TON_IOT.

A avaliação da arquitetura *ensemble* no *dataset* ToN_IoT oferece insights complementares aos resultados obtidos no CICIDS2017, permitindo validar a robustez da abordagem proposta em contextos de IoT e ambientes industriais. O ToN_IoT apresenta características distintas de tráfego, incluindo padrões de comunicação específicos de dispositivos IoT e vetores de ataque adaptados a esses ambientes, tornando essencial avaliar como as diferentes arquiteturas *ensemble* se comportam neste cenário operacional específico.

A Tabela 5 apresenta os resultados comparativos entre *Ensemble* HSAE (HSAE+PCA+One-Class SVM) e *Ensemble* VAE (VAE+PCA+One-Class SVM) no *dataset* ToN_IoT, utilizando a mesma metodologia experimental estabelecida nas seções anteriores. Esta análise permitirá identificar se os ganhos observados no CICIDS2017 se mantêm em ambientes IoT, além de avaliar a adaptabilidade da arquitetura *ensemble* a diferentes perfis de tráfego e tipos de ataques. As curvas ROC correspondentes são apresentadas na Figura 8, proporcionando visualização detalhada do comportamento discriminativo de cada arquitetura *ensemble* nos diferentes tipos de ataque.

Tabela 5 – Comparação de desempenho dos *Ensembles* para o conjunto de dados ToN_IoT.

Ataque	**Modelo**	**Precisão**	**FPR**	**FNR**	***Recall***	***F1-Score***	**AUC**
DDoS	Ens.VAE	75%	25,34%	**24,80%**	***75%***	*75%*	**0.83**
	Ens.HSAE	**97%**	**2,15%**	26,28%	*74%*	***84%***	0.82
DoS	Ens.VAE	78%	18,82%	32,49%	*68%*	*73%*	0.82
	Ens.HSAE	**95%**	**4,45%**	**10,10%**	***90%***	***93%***	**0.93**
Ransomware	Ens.VAE	63%	42,35%	28,73%	*71%*	*67%*	0.71
	Ens.HSAE	**94%**	**5,82%**	**5,40%**	***95%***	***94%***	**0.96**
Múltiplos Ataques	Ens.VAE	77%	41,65%	39,61%	*70%*	*74%*	0.70
	Ens.HSAE	**96%**	**7,38%**	**5,15%**	***95%***	***96%***	**0.95**

5.4.1 Análise Comparativa Detalhada

Ataques DDoS: O *Ensemble* HSAE demonstrou precisão excepcional de 97% comparado aos 75% do *Ensemble* VAE, representando um ganho substancial de 22 pontos percentuais. A taxa de falsos positivos foi drasticamente reduzida de 25,34% para 2,15%, evidenciando

controle superior de alarmes indevidos - aspecto crítico para ambientes IoT onde recursos computacionais são limitados. Entretanto, observa-se um *trade-off* no *recall*, que foi de 74% para o *Ensemble* HSAE versus 75% para o *Ensemble* VAE, uma diferença marginal de 1 ponto percentual. O *F1-score* do *Ensemble* HSAE atingiu 84% contra 75% do *Ensemble* VAE, enquanto a AUC foi ligeiramente inferior (0,82 vs 0,83).

Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 3: 85% precisão, 84% recall, 0,91 AUC), o *ensemble* apresenta ganhos significativos em precisão (+12 pontos percentuais) em contrapartida a uma redução no *recall* (84% $\rightarrow$ 74%, -10 pontos percentuais). A AUC apresenta redução de 0,91 para 0,82, sugerindo que para ataques DDoS no contexto IoT, o *ensemble* prioriza precisão extremamente alta em detrimento da sensibilidade geral.

DoS: O *Ensemble* HSAE alcançou 95% de precisão, superando significativamente os 78% do *Ensemble* VAE. A FPR foi reduzida de 18,82% para 4,45%, demonstrando controle excepcional de falsos positivos. O *recall* atingiu 90% versus 68% do *Ensemble* VAE, resultando em *F1-score* superior (93% vs 73%). A AUC evoluiu de 0,82 para 0,93, indicando capacidade discriminativa substancialmente superior.

Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 3: 91% precisão, 77% recall, 0,77 AUC), o *ensemble* apresenta melhorias consistentes em todas as métricas: +4 pontos percentuais em precisão, +13 pontos em *recall*, +8 pontos em *F1-score* e +0,16 na AUC. Este comportamento sugere que a arquitetura *ensemble* é particularmente eficaz para ataques DoS em ambientes IoT.

Ransomware: Representa o melhor desempenho absoluto do *Ensemble* HSAE, com 94% de precisão superando dramaticamente os 63% do *Ensemble* VAE. A FPR foi reduzida de 42,35% para 5,82%, uma melhoria excepcional de 36,53 pontos percentuais. O *recall* alcançou 95% versus 71% do *Ensemble* VAE, resultando em *F1-score* de 94% contra 67%. A AUC atingiu 0,96 comparado aos 0,71 do *Ensemble* VAE, representando um salto qualitativo na capacidade discriminativa.

Comparando o *Ensemble* HSAE com o HSAE isolado (Tabela 3: 73% precisão, 70% *recall*, 0,68 AUC), o *ensemble* apresenta os maiores ganhos observados no *dataset* ToN_IoT: +21 pontos percentuais em precisão, +25 pontos em *recall*, +24 pontos em *F1-score* e +0,28 na AUC. Esta melhoria substancial indica que a arquitetura *ensemble* é especialmente adequada para detecção de *ransomware* em ambientes IoT, onde este tipo de ataque representa uma ameaça crítica. Múltiplos Ataques: Robustez em Ambientes IoT Complexos A avaliação com múltiplos ataques no contexto IoT revelou aspectos fundamentais sobre a adaptabilidade dos

ensembles. O *Ensemble* HSAE demonstrou 96% de precisão contra 77% do *Ensemble* VAE, evidenciando capacidade superior de manter performance em ambientes IoT heterogêneos. A FPR foi reduzida drasticamente de 41,65% para 7,38%, enquanto a FNR apresentou melhoria de 39,61% para 5,15%. O *recall* atingiu 95% versus 70% do *Ensemble* VAE, resultando em *F1-score* de 96% contra 74%. A AUC evoluiu de 0,70 para 0,95, representando um salto qualitativo excepcional na capacidade discriminativa.

5.4.2 Múltiplos Ataques: Robustez em Cenários Complexos

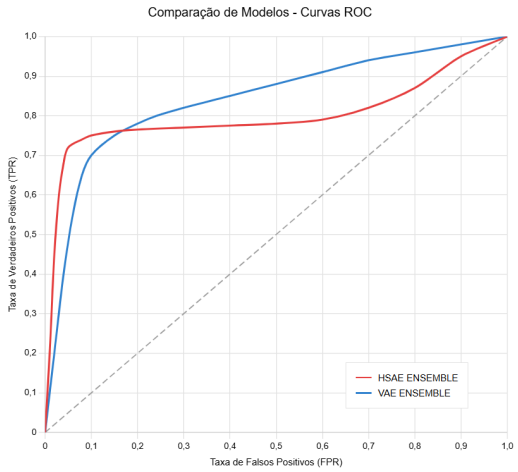
Comparando o *Ensemble* HSAE com o HSAE isolado em cenários de múltiplos ataques (Tabela 3: 94% precisão, 90% recall, 0,89 AUC), o *ensemble* apresenta ganhos consistentes: $+2$ pontos percentuais em precisão, $+5$ pontos em *recall* e $+0{,}06$ na AUC. Embora os ganhos sejam mais modestos que em outros tipos de ataque, demonstram robustez incremental em cenários complexos.

Análise Contextualizada para Ambientes IoT Os resultados no *dataset* ToN_IoT revelam características distintas em relação ao CICIDS2017, particularmente na magnitude dos ganhos proporcionados pelo *ensemble*. Para *ransomware*, onde o *ensemble* demonstra melhorias de mais de 20 pontos percentuais em múltiplas métricas, observa-se que a arquitetura *ensemble* consegue capturar padrões específicos deste tipo de ataque que são particularmente desafiadores no contexto IoT.

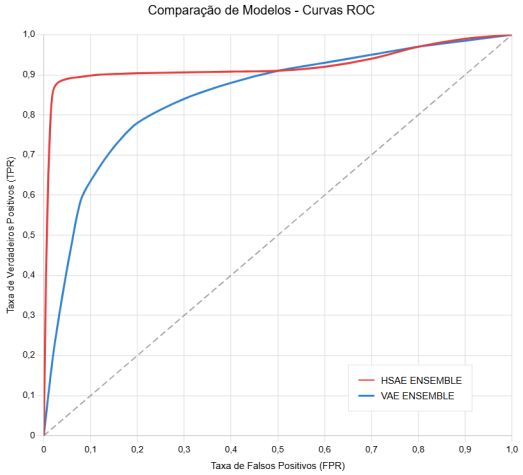
A redução consistente da FPR em todos os tipos de ataque (2,15% a 7,38% para o *Ensemble* HSAE versus 18,82% a 42,35% para o *Ensemble* VAE) é especialmente relevante em ambientes IoT, onde recursos computacionais limitados tornam custosa a investigação de falsos positivos. Esta característica posiciona o *Ensemble* HSAE como uma solução adequada para implantação em dispositivos com restrições de processamento.

5.4.3 Visualização Comparativa dos Resultados

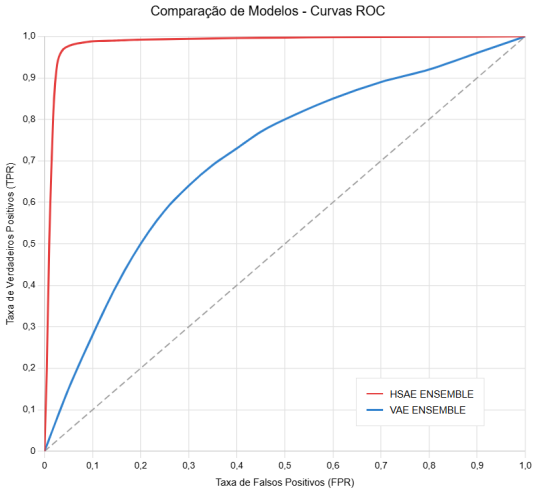
A análise das curvas ROC na Figura 17 corrobora estes achados, evidenciando a superioridade discriminativa do *Ensemble* HSAE, particularmente evidente nos ataques de *ransomware* e cenários de múltiplos ataques, onde as curvas demonstram maior área sob a curva e melhor separação entre classes.



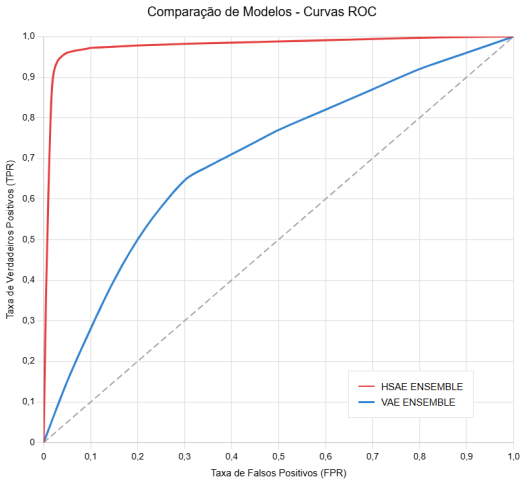
(a) Curva ROC para DDoS



(b) Curva ROC para DoS



(c) Curva ROC para Ramsoware



(d) Curva ROC para Múltiplos Ataques

Figura 17 – Comparação das curvas ROC para os *Ensembles* usando o *dataset* ToN_IoT.

5.4.4 Síntese Estratégica para IoT

Os resultados do *Ensemble* HSAE no *dataset* ToN_IoT estabelecem sua adequação superior para ambientes IoT, superando consistentemente o *Ensemble* VAE em todas as métricas avaliadas. A arquitetura demonstra particular eficácia em:

Detecção de *ransomware*: onde apresenta os maiores ganhos absolutos (+21% precisão, +25% recall sobre o modelo isolado), controle de falsos positivos: com FPR consistentemente baixa (2,15% a 7,38%), fundamental para ambientes com recursos limitados e robustez em múltiplos ataques: mantendo precisão >96% mesmo em cenários complexos

A escolha do *Ensemble* HSAE para ambientes IoT deve considerar os ganhos substantivos em precisão e controle de falsos positivos, que compensam a complexidade computacional

adicional em aplicações críticas onde a detecção confiável de ameaças é prioritária.

5.5 EFICIÊNCIA DE RECURSOS COMPUTACIONAIS E CONSUMO DE MEMÓRIA

Para a viabilidade de sistemas de detecção de intrusão em ambientes reais, especialmente sob restrições de recursos, é necessário avaliarmos a eficiência de memória dos modelos de sistema de detecção de anomalias. Para isso, comparamos nosso *ensemble* HSAE com o *framework* proposto por (SOLTANI et al., 2023), adotando a mesma métrica utilizada pelos autores — o tamanho do modelo carregado em memória ("*model size (in memory)*"), que considera exclusivamente os parâmetros do modelo, desconsiderando dados de entrada, bibliotecas externas e *overhead* do sistema.

O *framework* proposto em (SOLTANI et al., 2023) foi selecionado como base de comparação nesta análise por representar a única obra, dentre os seis estudos analisados na seção de trabalhos relacionados, que explicitamente reporta o consumo de memória do modelo como uma de suas métricas avaliativas. Essa escolha justifica-se não apenas pela relevância do *framework* proposto pelos autores no contexto de detecção de ataques *zero-day*, mas também pela possibilidade de estabelecer uma comparação objetiva com base no tamanho do modelo em memória ("*model size*"), conforme definido no próprio artigo. Assim, a comparação realizada nesta seção visa destacar, de forma justa e técnica, os ganhos obtidos em termos de eficiência computacional pelo *ensemble* da proposta HSAE em relação aos métodos de referência.

Embora a métrica de medição seja equivalente, é importante ressaltar uma diferença fundamental entre as abordagens. Enquanto os autores de (SOLTANI et al., 2023) utilizaram o *framework* Deep Intrusion Detection (DID), que converte dados de rede brutos (formato Packet Capture (PCAP)) em vetores de 20.000 dimensões (200 bytes $\times$ 100 pacotes), nossa abordagem opera diretamente sobre dados tabulares (formato Comma-Separated Values (CSV)) com 78 *features* originais do *dataset* CICIDS2017. O objetivo é quantificar o impacto estrutural e arquitetural de cada método sobre o consumo de memória do modelo.

5.5.1 Resultados Comparativos e Análise Arquitetural

Na Fase 1 do *framework* adaptativo proposto por Soltani et al. (2023) — a etapa de *open set recognition*, responsável por detectar amostras desconhecidas enquanto identifica

corretamente as conhecidas — os autores comparam cinco abordagens para medir desempenho e custo computacional, buscando o melhor equilíbrio entre precisão e viabilidade prática. A Tabela 6 sintetiza o tamanho do modelo na memória de cada método considerado por Soltani et al. (2023) e serve aqui como base para a nossa comparação: OpenMax (BENDALE; BOULT, 2016), 1,5 GB, baseado em CNN com recalibração por distribuições de Weibull sobre o vetor de ativações médias; Deep Open Classification (DOC) (SHU; XU; LIU, 2017), 1,5 GB, que substitui *softmax* por camadas 1-vs-rest com decisão por limiar; Classification-Reconstruction Learning for Open-Set Recognition (CROSR) (YOSHIHASHI et al., 2019), 4,5 GB, que concatena a representação Deep Hierarchical Representation Network (DHRNet) à do classificador e aplica OpenMax sobre essa concatenação, envolvendo treino em duas etapas e as propostas dos próprios autores, DOC++ e AutoSVM (SOLTANI et al., 2023), com 1,5 GB e 4,5 GB, respectivamente. No DOC++ os autores ensinam explicitamente, ainda no treino, a existência de classes desconhecidas por meio de amostras suplementares, mantendo decisão por limiar, enquanto no AutoSVM um *Stacked Autoencoder* reduz a dimensionalidade e quatro One-Class SVMs (um por classe conhecida) fazem a rejeição do que não pertence a nenhuma classe. Nos resultados reportados, o DOC++ aparece como o método de melhor desempenho na Fase 1.

Tabela 6 – Comparação de Consumo de Memória por Componente

Framework	**Método**	**Tamanho do Modelo**	**Componentes Principais**
Bendale & Boult (2016)	OpenMax	1.5 GB	CNN + Weibull distributions
Shu et al. (2017)	DOC	1.5 GB	CNN + 1-vs-rest layers
Yoshihashi et al. (2019)	CROSR	4.5 GB	DHRNet + Classificador CNN com OpenMax aplicado na concatenação das representações
Soltani et al. (2023)	DOC++	1.5 GB	CNN + 1-vs-rest layers
Soltani et al. (2023)	AutoSVM	4.5 GB	Stacked AE + 4xOne-Class SVMs
Ensemble HSAE	Pipeline Integrado	7,89 MB	HSAE + PCA + One-Class SVM

Com essa linha de base, comparamos diretamente o nosso *ensemble* HSAE ao DOC++, que apresentou o melhor desempenho. O tamanho do *Ensemble* HSAE medido em execução é de 7,89 MB (CICIDS2017 em cenário multiclasse de ataques), o que permite comparação direta com o cenário multiclasse do DOC++. A diferença de consumo de memória, 99,8%

menor em relação a CROSR/AutoSVM e 99,5% menor em relação ao próprio DOC++, decorre de quatro fatores arquiteturais do HSAE: (i) entrada e dimensionalidade otimizadas, pois enquanto Soltani et al. (2023) utilizam o DID com 20.000 dimensões extraídas de PCAP, operamos diretamente sobre 78 atributos estatísticos do CSV do CICIDS2017, reduzindo a carga de entrada; (ii) arquitetura híbrida unificada, integrando reconstrução e classificação em uma única estrutura com função de perda combinada, ao passo que arranjos como o CROSR exigem componentes adicionais (DHRNet + classificador + OpenMax); (iii) redução dimensional via PCA com 95% de variância antes do One-Class SVM, diminuindo a complexidade do classificador frente a alternativas baseadas em múltiplos SVMs de alta dimensionalidade; e (iv) eliminação de redundâncias, já que, enquanto Soltani et al. (2023) implementam cinco métodos distintos para a Fase 1, o *Ensemble* HSAE adota um *pipeline* único e enxuto, com menos parâmetros e manutenção simplificada.

5.5.2 Consumo Total de Sistema e Implicações Práticas

Embora as abordagens utilizem entradas distintas (PCAP vs. CSV), consideramos que a métrica de comparação escolhida — o consumo do modelo em memória — pode oferecer uma base comparativa útil. A escolha por dados tabulares busca equilibrar eficiência computacional com aplicabilidade prática, procurando manter a capacidade de detecção dentro de limitações aceitáveis.

Adicionalmente à medição do *ensemble* HSAE isolado, monitoramos também o consumo total de memória do processo Python, utilizando o indicador Resident Set Size (RSS) durante a execução do teste. Esse valor representa a quantidade total de memória residente ocupada em Random Access Memory (RAM) durante a execução do sistema completo, incluindo os dados carregados em memória, as bibliotecas Python e o *overhead* do interpretador. O valor registrado foi de 3.547,51 MB, que corresponde ao pico de uso durante a inferência sobre múltiplos tipos de ataques. Ainda assim, esse total completo permanece aproximadamente 27% inferior ao tamanho isolado do modelo AutoSVM reportado por e (SOLTANI et al., 2023), que é de 4.500 MB apenas para os parâmetros dos modelos, sem considerar o *overhead* adicional do sistema.

A eficiência de memória obtida pelo *ensemble* HSAE apresenta implicações práticas significativas. O modelo de 7,89 MB permite *deployment* em dispositivos com recursos limitados, incluindo *edge computing* e IoT *gateways*, contextos onde os 4,5 GB do *framework* completo

de Soltani seria proibitivo. O menor consumo de memória traduz-se diretamente em redução de custos de infraestrutura em ambientes cloud e permite execução simultânea de múltiplas instâncias do modelo em um único nó, aumentando a vazão do sistema.

Os resultados demonstram que o *ensemble* HSAE alcança eficiência de memória superior através de design arquitetural otimizado, mantendo performance de detecção competitiva. A redução de 99,8% no consumo de memória, combinada com a eliminação da necessidade de intervenção manual (*clustering* e rotulagem por especialistas), posiciona o *ensemble* HSAE como uma alternativa mais prática e escalável para *deployment* em ambientes de produção. Esse resultado reforça que, mesmo considerando todos os fatores do ambiente de execução real, o *ensemble* da proposta HSAE apresenta uma eficiência global superior, questionando a necessidade de *frameworks* multi-fase complexos para problemas de detecção de anomalias em redes.

6 CONCLUSÃO E SUGESTÕES

6.1 CONCLUSÃO

Este trabalho apresentou o HSAE, uma arquitetura híbrida não supervisionada para detecção de ataques *Zero-Day*, juntamente com sua extensão baseada em *ensemble*. A solução proposta combina o erro de reconstrução com mecanismos de pontuação híbrida e limiares de decisão dinâmicos otimizados via *Equal Error Rate*. Esta abordagem demonstrou superioridade em relação ao modelo de referência baseado em *Variational Autoencoder* (VAE). A arquitetura *ensemble*, que integra HSAE, PCA e One-Class SVM, apresentou robustez e capacidade de generalização em diferentes ambientes de rede. O método manteve altos níveis de precisão e baixas taxas de falsos positivos, o que demonstra sua eficácia na identificação de tráfego anômalo mesmo na ausência de dados maliciosos previamente rotulados no processo de treinamento.

A avaliação empírica nos conjuntos de dados CICIDS2017 e ToN_IoT confirmou a superioridade de ambas as propostas em métricas-chave. O modelo *ensemble* apresentou um bom desempenho, alcançando 94% de precisão e 96% de AUC na detecção de *Ransomware* no conjunto ToN_IoT. Em cenários de múltiplos ataques, a precisão atingiu 96%. Esses resultados evidenciam a eficácia das inovações introduzidas, como o sistema de pontuação híbrida e o limiar dinâmico, além de sua capacidade de endereçar limitações documentadas na literatura.

A redução na taxa de falsos positivos ocorreu em quase todos os cenários testados. O desempenho consistente acima de 90% na maioria das configurações valida a robustez da arquitetura em ambientes heterogêneos. Estes incluem tanto redes corporativas tradicionais quanto ambientes modernos de IoT e IIoT. Adicionalmente, o trabalho introduz uma metodologia de pré-processamento transparente que contribui para a mitigação de vieses e falhas comuns em experimentos com aprendizado não supervisionado. Essa característica, aliada à leveza computacional da arquitetura, torna o HSAE uma alternativa promissora para aplicações em tempo real e com restrições de recursos.

O sistema desenvolvido pode ser aplicado em soluções de segurança de rede para a detecção proativa de uma vasta gama de ameaças. Quando integrado com recursos computacionais adequados, ele aprimora as operações de segurança como ferramenta complementar para monitoramento e resposta a incidentes. Sua eficiência o torna particularmente valioso para dispositivos com recursos limitados. É fundamental considerar a escolha entre o modelo HSAE

isolado e sua versão *ensemble* para garantir uma implementação confiável. Essa decisão permite que a solução se ajuste aos requisitos específicos de latência, precisão e custo computacional de diferentes cenários operacionais. Os códigos dos modelos, contendo as principais funções, está disponível na plataforma Github[1].

6.2 CONTRIBUIÇÕES DA PESQUISA

Abaixo foram listadas as contribuições realizadas por esta pesquisa:

- **P**roposição de uma abordagem híbrida não supervisionada baseada em *autoencoder*, fundamentada na detecção comportamental e na combinação multi-critérios de erro de reconstrução e pontuação direta de anomalias, com uso de função de perda híbrida e *score* ponderado, superando limitações de métodos tradicionais baseados em assinaturas;
- **D**esenvolvimento de uma arquitetura *ensemble* sequencial leve e adequada a dispositivos com restrição de recursos, integrando aprendizado de representação, redução dimensional e modelagem de fronteiras de normalidade, com o objetivo de aprimorar a detecção de ataques *zero-day* em ambientes IoT e IIoT.

6.3 TRABALHOS FUTUROS

Para investigações futuras, a avaliação do desempenho do HSAE em ambientes reais com restrições computacionais pode ser expandida para diferentes contextos operacionais, como *gateways* IoT, sensores industriais e dispositivos embarcados, sendo possível implementar módulos de detecção em tempo real acoplados ao HSAE com estudos aprofundados de latência e resposta em cenários operacionais diversos. Expansões do modelo para novos conjuntos de dados também podem ser propostas, abrangendo redes IoT médicas (Internet of Medical Things (IoMT)), sistemas veiculares (Vehicular Ad-hoc Network (VANETs)) e ambientes de cidades inteligentes, permitindo a integração de mecanismos automáticos de resposta que incluem aplicação de políticas de mitigação, geração de alertas ou isolamento de nós comprometidos. Adaptações do modelo para cenários de aprendizado contínuo podem ser desenvolvidas, possibilitando atualizações incrementais e maior resiliência frente a padrões de tráfego em constante

[1] Disponível em: <https://github.com/fabianoinfosec/Dissertation_Codes>

evolução, o que representa um avanço significativo na capacidade de detecção e resposta a ameaças em tempo real.

REFERÊNCIAS

ABBAS, S.; NASER, W.; KADHIM, A. Subject review: Intrusion detection system (ids) and intrusion prevention system (ips). *Global Journal of Engineering and Technology Advances*, v. 2, n. 14, p. 155–158, 2023.

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, v. 22, n. 5, p. 1125–1162, 2023.

AHMAD, R.; ALSMADI, I.; ALHAMDANI, W.; TAWALBEH, L. Zero-day attack detection: a systematic literature review. *Artificial Intelligence Review*, Springer, v. 56, n. 10, p. 10733–10811, 2023.

AHMED, M.; SERAJ, R.; ISLAM, S. M. S. The k-means algorithm: A comprehensive survey and performance evaluation. *Electronics*, MDPI, v. 9, n. 8, p. 1295, 2020.

ALKASASSBEH, M.; BADDAR, S. A.-H. Intrusion detection systems: A state-of-the-art taxonomy and survey. *Arabian Journal for Science and Engineering*, Springer, v. 48, n. 8, p. 10021–10064, 2023.

BANK, D.; KOENIGSTEIN, N.; GIRYES, R. Autoencoders. *Machine learning for data science handbook: data mining and knowledge discovery handbook*, Springer, p. 353–374, 2023.

BEAMAN, C.; BARKWORTH, A.; AKANDE, T. D.; HAKAK, S.; KHAN, M. K. Ransomware: Recent advances, analysis, challenges and future research directions. *Computers & security*, Elsevier, v. 111, p. 102490, 2021.

BENDALE, A.; BOULT, T. E. Towards open set deep networks. In: *Proceedings of the IEEE conference on computer vision and pattern recognition*. [S.l.: s.n.], 2016. p. 1563–1572.

BERAHMAND, K.; DANESHFAR, F.; SALEHI, E. S.; LI, Y.; XU, Y. Autoencoders and their applications in machine learning: a survey. *Artificial intelligence review*, Springer, v. 57, n. 2, p. 28, 2024.

BILGE, L.; DUMITRAŞ, T. Before we knew it: an empirical study of zero-day attacks in the real world. In: *Proceedings of the 2012 ACM conference on Computer and communications security*. [S.l.: s.n.], 2012. p. 833–844.

BUDIARTO, E. H.; PERMANASARI, A. E.; FAUZIATI, S. Unsupervised anomaly detection using k-means, local outlier factor and one class svm. In: IEEE. *2019 5th international conference on science and technology (ICST)*. [S.l.], 2019. v. 1, p. 1–5.

CHEN, Q.; BRIDGES, R. A. Automated behavioral analysis of malware: A case study of wannacry ransomware. In: IEEE. *2017 16th IEEE International Conference on machine learning and applications (ICMLA)*. [S.l.], 2017. p. 454–460.

CHENG, J.-M.; WANG, H.-C. A method of estimating the equal error rate for automatic speaker verification. In: IEEE. *2004 International Symposium on Chinese Spoken Language Processing*. [S.l.], 2004. p. 285–288.

CRAIGEN, D.; DIAKUN-THIBAULT, N.; PURSE, R. Defining cybersecurity. *Technology innovation management review*, v. 4, n. 10, 2014.

DEOGIRIKAR, J.; VIDHATE, A. Security attacks in iot: A survey. In: IEEE. *2017 International conference on I-SMAC (IoT in social, mobile, analytics and cloud)(I-SMAC)*. [S.l.], 2017. p. 32–37.

ELOUARDI, S.; MOTII, A.; JOUHARI, M.; AMADOU, A. N. H.; HEDABOU, M. A survey on hybrid-cnn and llms for intrusion detection systems: Recent iot datasets. *IEEE Access*, IEEE, 2024.

FARIZI, W. S. A.; HIDAYAH, I.; RIZAL, M. N. Isolation forest based anomaly detection: A systematic literature review. In: IEEE. *2021 8th International Conference on Information Technology, Computer and Electrical Engineering (ICITACEE)*. [S.l.], 2021. p. 118–122.

GANDOTRA, E.; BANSAL, D.; SOFAT, S. Zero-day malware detection. In: IEEE. *2016 sixth international symposium on embedded computing and system design (ISED)*. [S.l.], 2016. p. 171–175.

GARCIA-TEODORO, P.; DIAZ-VERDEJO, J.; MACIÁ-FERNÁNDEZ, G.; VÁZQUEZ, E. Anomaly-based network intrusion detection: Techniques, systems and challenges. *computers & security*, Elsevier, v. 28, n. 1-2, p. 18–28, 2009.

GHARIB, A.; SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. An evaluation framework for intrusion detection dataset. In: IEEE. *2016 International conference on information science and security (ICISS)*. [S.l.], 2016. p. 1–6.

Google Developers. *Curso Rápido de Aprendizado de Máquina - Curva ROC e AUC*. 2024. Acessado em: 23 jul. 2025. Disponível em: <https://developers.google.com/machine-learning/crash-course/classification/roc-and-auc?hl=pt-br>.

GUO, Y. A review of machine learning-based zero-day attack detection: Challenges and future directions. *Computer communications*, Elsevier, v. 198, p. 175–185, 2023.

HALVORSEN, J.; IZURIETA, C.; CAI, H.; GEBREMEDHIN, A. Applying generative machine learning to intrusion detection: A systematic mapping study and review. *ACM Computing Surveys*, ACM New York, NY, v. 56, n. 10, p. 1–33, 2024.

HOQUE, N.; BHUYAN, M. H.; BAISHYA, R. C.; BHATTACHARYYA, D. K.; KALITA, J. K. Network attacks: Taxonomy, tools and systems. *Journal of Network and Computer Applications*, Elsevier, v. 40, p. 307–324, 2014.

JAVAHERI, D.; GORGIN, S.; LEE, J.-A.; MASDARI, M. Fuzzy logic-based ddos attacks and network traffic anomaly detection methods: Classification, overview, and future perspectives. *Information Sciences*, Elsevier, v. 626, p. 315–338, 2023.

KUMAR, S. S.; SELVI, M.; KANNAN, A. A comprehensive survey on machine learning-based intrusion detection systems for secure communication in internet of things. *Computational Intelligence and Neuroscience*, Wiley Online Library, v. 2023, n. 1, p. 8981988, 2023.

LARSON, U. E.; NILSSON, D. K.; JONSSON, E.; LINDSKOG, S. Using system call information to reveal hidden attack manifestations. In: IEEE. *2009 Proceedings of the 1st International Workshop on Security and Communication Networks*. [S.l.], 2009. p. 1–8.

LATHA, S.; PRAKASH, S. J. A survey on network attacks and intrusion detection systems. In: IEEE. *2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS)*. [S.l.], 2017. p. 1–7.

LIU, L.; VEL, O. D.; HAN, Q.-L.; ZHANG, J.; XIANG, Y. Detecting and preventing cyber insider threats: A survey. *IEEE Communications Surveys & Tutorials*, IEEE, v. 20, n. 2, p. 1397–1417, 2018.

LONE, A. N.; MUSTAJAB, S.; ALAM, M. A comprehensive study on cybersecurity challenges and opportunities in the iot world. *Security and Privacy*, Wiley Online Library, v. 6, n. 6, p. e318, 2023.

LU, H.; ZHAO, Y.; SONG, Y.; YANG, Y.; HE, G.; YU, H.; REN, Y. A transfer learning-based intrusion detection system for zero-day attack in communication-based train control system. *Cluster Computing*, Springer, v. 27, n. 6, p. 8477–8492, 2024.

MALLIGA, S.; NANDHINI, P.; KOGILAVANI, S. V. A comprehensive review of deep learning techniques for the detection of (distributed) denial of service attacks. *Information Technology and Control*, v. 51, n. 1, p. 180–215, 2022.

MBONA, I.; ELOFF, J. H. Detecting zero-day intrusion attacks using semi-supervised machine learning approaches. *Ieee Access*, IEEE, v. 10, p. 69822–69838, 2022.

MINHAS, M. R.; SHAFI, Q.; KHAN, S. A.; AHMAD, T.; ULLAH, S.; BURIRO, A.; YAQUB, M. A. F-osfa: A fog level generalizable solution for zero-day ddos attacks detection. *IEEE Access*, IEEE, 2025.

MOUSTAFA, N. *A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustain. Cities Soc. 72, 102994 (2021)*. 2021.

NARAYAN, S. The generalized sigmoid activation function: Competitive supervised learning. *Information sciences*, Elsevier, v. 99, n. 1-2, p. 69–82, 1997.

NARKHEDE, S. Understanding auc-roc curve. *Towards data science*, v. 26, n. 1, p. 220–227, 2018.

NEIRA, A. B. D.; KANTARCI, B.; NOGUEIRA, M. Distributed denial of service attack prediction: Challenges, open issues and opportunities. *Computer Networks*, Elsevier, v. 222, p. 109553, 2023.

NEUPANE, S.; ABLES, J.; ANDERSON, W.; MITTAL, S.; RAHIMI, S.; BANICESCU, I.; SEALE, M. Explainable intrusion detection systems (x-ids): A survey of current methods, challenges, and opportunities. *IEEE Access*, IEEE, v. 10, p. 112392–112415, 2022.

NKONGOLO, M.; TOKMAK, M. Zero-day threats detection for critical infrastructures. In: SPRINGER. *Annual Conference of South African Institute of Computer Scientists and Information Technologists*. [S.l.], 2023. p. 32–47.

PANG, G.; SHEN, C.; CAO, L.; HENGEL, A. V. D. Deep learning for anomaly detection: A review. *ACM computing surveys (CSUR)*, ACM New York, NY, USA, v. 54, n. 2, p. 1–38, 2021.

PRABHU, S.; THOMPSON, N. A primer on insider threats in cybersecurity. *Information Security Journal: A Global Perspective*, Taylor & Francis, v. 31, n. 5, p. 602–611, 2022.

PRATIWI, H.; WINDARTO, A. P.; SUSLIANSYAH, S.; ARIA, R. R.; SUSILOWATI, S.; RAHAYU, L. K.; FITRIANI, Y.; MERDEKAWATI, A.; RAHADJENG, I. R. Sigmoid activation function in selecting the best model of artificial neural networks. In: IOP PUBLISHING. *Journal of physics: conference series*. [S.l.], 2020. v. 1471, n. 1, p. 012010.

RAINIO, O.; TEUHO, J.; KLÉN, R. Evaluation metrics and statistical tests for machine learning. *Scientific Reports*, Nature Publishing Group UK London, v. 14, n. 1, p. 6086, 2024.

RAZAULLA, S.; FACHKHA, C.; MARKARIAN, C.; GAWANMEH, A.; MANSOOR, W.; FUNG, B. C.; ASSI, C. The age of ransomware: A survey on the evolution, taxonomy, and research directions. *IEEE Access*, IEEE, v. 11, p. 40698–40723, 2023.

ROUMANI, Y. Patching zero-day vulnerabilities: an empirical analysis. *Journal of Cybersecurity*, Oxford University Press, v. 7, n. 1, p. tyab023, 2021.

SAMARIYA, D.; THAKKAR, A. A comprehensive survey of anomaly detection algorithms. *Annals of Data Science*, Springer, v. 10, n. 3, p. 829–850, 2023.

SAMONAS, S.; COSS, D. The cia strikes back: Redefining confidentiality, integrity and availability in security. *Journal of Information System Security*, v. 10, n. 3, 2014.

SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. et al. Toward generating a new intrusion detection dataset and intrusion traffic characterization. *ICISSp*, v. 1, n. 2018, p. 108–116, 2018.

SHENG, C.; ZHOU, W.; HAN, Q.-L.; MA, W.; ZHU, X.; WEN, S.; XIANG, Y. Network traffic fingerprinting for iiot device identification: A survey. *IEEE Transactions on Industrial Informatics*, IEEE, 2025.

SHOREY, T.; SUBBAIAH, D.; GOYAL, A.; SAKXENA, A.; MISHRA, A. K. Performance comparison and analysis of slowloris, goldeneye and xerxes ddos attack tools. In: IEEE. *2018 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*. [S.l.], 2018. p. 318–322.

SHU, L.; XU, H.; LIU, B. Doc: Deep open classification of text documents. *arXiv preprint arXiv:1709.08716*, 2017.

SOLTANI, M.; OUSAT, B.; SIAVOSHANI, M. J.; JAHANGIR, A. H. An adaptable deep learning-based intrusion detection system to zero-day attacks. *Journal of Information Security and Applications*, Elsevier, v. 76, p. 103516, 2023.

VERIZON. *2024 Data Breach Investigations Report*. [S.l.], 2024. Disponível em: <https://www.verizon.com/business/resources/reports/2024-dbir-data-breach-investigations-report.pdf>.

VISHWAKARMA, R.; JAIN, A. K. A survey of ddos attacking techniques and defence mechanisms in the iot network. *Telecommunication systems*, Springer, v. 73, n. 1, p. 3–25, 2020.

WAGNER, D.; SOTO, P. Mimicry attacks on host-based intrusion detection systems. In: *Proceedings of the 9th ACM Conference on Computer and Communications Security*. [S.l.: s.n.], 2002. p. 255–264.

YANG, Z.; LIU, X.; LI, T.; WU, D.; WANG, J.; ZHAO, Y.; HAN, H. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. *Computers & Security*, Elsevier, v. 116, p. 102675, 2022.

YOSHIHASHI, R.; SHAO, W.; KAWAKAMI, R.; YOU, S.; IIDA, M.; NAEMURA, T. Classification-reconstruction learning for open-set recognition. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. [S.l.: s.n.], 2019. p. 4016–4025.

YUAN, S.; WU, X. Deep learning for insider threat detection: Review, challenges and opportunities. *Computers & Security*, Elsevier, v. 104, p. 102221, 2021.

ZAHOORA, U.; RAJARAJAN, M.; PAN, Z.; KHAN, A. Zero-day ransomware attack detection using deep contractive autoencoder and voting based ensemble classifier. *Applied Intelligence*, Springer, v. 52, n. 12, p. 13941–13960, 2022.

ZAVRAK, S.; ISKEFIYELI, M. Anomaly-based intrusion detection from network flow features using variational autoencoder. *IEEe Access*, IEEE, v. 8, p. 108346–108358, 2020.