

CyberEye: Sistema de Detecção de Intrusão em Redes com Integração de Inteligência Artificial

Ígor Moreno Boenavides¹, Jardel Batista Gonçalves², Matheus Lorenzato Braga¹

¹Instituto Federal Catarinense - Campus Sombrio - Sombrio, SC - Brasil

²Universidade Federal de Santa Catarina - Campus Araranguá - Araranguá, SC - Brasil

igor.boenavides@proton.me, jardel.goncalves@ifc.edu.br,
matheus.braga@ifc.edu.br

Abstract. *This work proposes the development of CyberEye, a Network Intrusion Detection System (NIDS) based on the Unsupervised Machine Learning algorithm Isolation Forest. The system captures and extracts 15 network traffic features in real time using the Scapy framework, training the model on benign traffic from the CIC-IDS-2017 dataset. As a proof-of-concept evaluation, tests in a controlled virtual environment demonstrated 100% detection rate for volumetric flood attacks and port scanning across 37 three-second capture windows, with zero false positives on idle benign traffic. The open-source proposal aims to broaden access to contemporary, cost-effective security tools for small and medium-sized organizations.*

Resumo. *Este trabalho propõe o desenvolvimento do CyberEye, um Sistema de Detecção de Intrusão em Redes (NIDS) baseado no algoritmo de Aprendizado de Máquina Não Supervisionado Isolation Forest. O sistema captura e extrai 15 características do tráfego de rede em tempo real utilizando o framework Scapy, treinando o modelo com tráfego benigno do dataset CIC-IDS-2017. Em avaliação de prova de conceito em ambiente virtual controlado, testes em 37 janelas de captura de 3 segundos demonstraram 100% de detecção para ataques volumétricos de inundação e varredura de portas, com zero falsos positivos sobre tráfego benigno ocioso. A proposta de código aberto visa ampliar o acesso a ferramentas de segurança contemporâneas e de baixo custo para pequenas e médias organizações.*

1. Introdução

As infraestruturas digitais se tornaram um pilar fundamental para o funcionamento da maioria das organizações; juntamente do desenvolvimento dessas tecnologias, acompanha a sofisticação das ameaças cibernéticas. Os sistemas de detecção tradicionais têm se tornado cada vez mais inadequados frente a esse cenário, há um consenso de que técnicas de *machine learning* (ML) e *deep learning* superam os *Intrusion Detection Systems* (IDS) tradicionais em precisão e capacidade de lidar com tráfego moderno, sobretudo usando *datasets* recentes (Ghadermazi, 2024). Além disso, ataques de *ransomware* e invasões em redes corporativas geram prejuízos enormes todos os anos, e as principais soluções robustas são proprietárias, com taxas de licenciamento restritivas para organizações de menor escala (Fernando, 2020; Rawindaran, 2021).

Diante deste contexto, este trabalho delimita-se ao desenvolvimento de um *Network Intrusion Detection System* (NIDS) com integração de ML, buscando responder: como modelos de detecção baseados em IA podem auxiliar na detecção de intrusão em redes de computadores? O objetivo geral é desenvolver um NIDS baseado no *Isolation Forest* (escolhido por sua capacidade de operar sem dados rotulados de ataque, eficiência computacional com grandes volumes de dados e robustez frente a

ruídos) capaz de detectar anomalias de tráfego em tempo real. Os objetivos específicos são: capturar e extrair 15 *features* de tráfego usando o *framework* Scapy; treinar o modelo com tráfego benigno do *dataset* CIC-IDS-2017; implementar *pipeline* de detecção em tempo real com geração de alertas em CSV; e avaliar o desempenho contra ataques simulados (ICMP Flood, SYN Flood, Port Scan).

A proposta, baseada em ML e filosofia *open source*, oferece uma alternativa financeiramente viável e contribui academicamente ao testar a eficácia de modelos de detecção em tempo real (Monteiro, 2026). Esta pesquisa se apoia em estudos recentes que relacionam cibersegurança e ML.

2. Trabalhos Relacionados

O algoritmo Isolation Forest foi proposto por Liu, Ting e Zhou (2008), que demonstraram sua eficiência para detecção de anomalias em grandes volumes de dados com baixo custo computacional. O *dataset* CIC-IDS-2017, utilizado para o treinamento do modelo com tráfego benigno, foi desenvolvido por Sharafaldin, Habibi Lashkari e Ghorbani (2018), que sistematizaram a geração de tráfego realista para avaliação de NIDS. Vanin et al. (2022) avaliaram sistematicamente IDS potencializados por algoritmos de ML, concluindo que abordagens baseadas em anomalia superam os sistemas baseados em assinatura para ataques não catalogados. Moamin et al. (2025) realizaram levantamento abrangente sobre aplicações de IA na detecção de malware e intrusão, destacando técnicas não supervisionadas para cenários sem dados rotulados. Shen et al. (2022) revisaram técnicas de ML para análise de tráfego cifrado, reforçando a sua relevância no contexto de redes modernas.

3. Metodologia

Esta pesquisa adota a Design Science Research (DSR) como método de condução (Dresch, Lacerda e Antunes Júnior, 2015). A DSR orienta pesquisas cujo objetivo é projetar e desenvolver artefatos que solucionem problemas práticos de forma rigorosa e cientificamente fundamentada. A seguir, descrevem-se as etapas percorridas.

3.1 Identificação do Problema

O problema identificado é a crescente inadequação dos IDS tradicionais frente à diversidade e à sofisticação das ameaças cibernéticas mais atuais. Esses sistemas são incapazes de detectar ataques desconhecidos ou variações de ataques previamente não catalogados (*zero-day*), além de exigirem atualizações constantes de suas bases de assinaturas.

3.2 Conscientização do Problema

A conscientização do problema foi conduzida por meio de revisão bibliográfica da literatura de segurança de redes, IDS e ML. Foram revisados estudos que apontam o crescimento dos ataques cibernéticos, as limitações dos IDS baseados em assinatura e o estado da arte em abordagens de detecção baseadas em anomalia. Essa etapa evidenciou que abordagens não supervisionadas, por não dependerem de dados rotulados de ataque, são especialmente adequadas para ambientes de rede dinâmicos com ameaças inéditas (Alzahrani e Alenazi, 2021).

3.3 Identificação, Proposição e Projeto do Artefato

A partir da conscientização do problema, identificou-se que o artefato adequado é um sistema computacional do tipo NIDS. O algoritmo Isolation Forest foi selecionado com base na literatura revisada, que o aponta como eficiente para detecção de anomalias em grandes volumes de dados, sem necessidade de rótulos de ataque para o treinamento. O artefato proposto foi denominado CyberEye¹.

O CyberEye foi projetado como um *pipeline* modular em Python com cinco módulos: `analyzer.py`, `train_model.py`, `alert_system.py`, `logger.py` e `evaluate.py`. O projeto definiu 15 características a extrair do tráfego em janelas de tempo, cobrindo métricas de volume, protocolo, flags TCP e portas de destino.

Os 15 atributos extraídos de cada janela incluem contagens de pacotes e *bytes*, taxas de tráfego, proporções por protocolo (TCP, UDP, ICMP), *flags* TCP (SYN, FIN, ACK), número de portas de destino únicas, e estatísticas de tamanho de pacote (média e desvio padrão).

3.4 Desenvolvimento do Artefato

O módulo `analyzer.py` captura pacotes assincronamente em janelas de 3 segundos, calculando o vetor de 15 *features*. O módulo `train_model.py` carrega 2 milhões de registros benignos do CIC-IDS-2017, pré-processa via `RobustScaler` e treina o Isolation Forest com 200 estimadores. O limiar foi definido no percentil 2,5% da distribuição de *scores* gerados pelo modelo sobre o próprio tráfego benigno do *dataset*, ou seja, o valor abaixo do qual se encontram apenas as 2,5% de janelas mais atípicas do tráfego normal. O Isolation Forest atribui *scores* próximos de zero ou positivos a padrões benignos e valores crescentemente negativos a anomalias; o percentil 2,5% resultou em *threshold* de -0,1203, abaixo do qual qualquer janela é classificada como anomalia e gera alerta.

3.5 Avaliação do Artefato

A avaliação do CyberEye foi conduzida em um ambiente de testes controlado, composto por: um *notebook host* com Fedora Linux (16 GB de memória RAM, CPU de 8 núcleos), executando o CyberEye em modo de monitoramento, e uma máquina virtual com Kali Linux (2 GB de memória RAM, CPU de 2 núcleos), conectada via rede *Host-Only* do VirtualBox, utilizada como origem dos ataques simulados. Três vetores de ataque foram simulados via ferramentas de linha de comando: *ICMP Flood* (`hping3 --icmp --flood`), *SYN Flood* (`hping3 --syn -p 80 --flood`) e *Port Scan* (`nmap -sS -p 1-5000 -T4`). Durante cada simulação, o CyberEye operou em tempo real, classificando cada janela de 3 segundos com base no *threshold* definido. Os critérios de avaliação foram a taxa de detecção por cenário e a taxa de falsos positivos sobre tráfego benigno.

¹ Código disponível em: <https://github.com/igor-boenavides/cyber-eye>.

4. Resultados e Discussões

A validação do CyberEye ocorreu em duas etapas: avaliação do perfil de tráfego e definição do limiar de detecção e testes expandidos de eficácia contra vetores de ataque simulados.

4.1 Análise de Atributos e Limiar

A extração contínua de 15 atributos via Scapy em janelas de 3 segundos permitiu caracterizar o perfil estatístico do tráfego. A Figura 1 apresenta a distribuição dos *scores* gerados durante as capturas. O tráfego benigno concentrou-se predominantemente em pontuações superiores a -0.10, enquanto os cenários maliciosos apresentaram desvios acentuados, agrupando-se em faixas inferiores (entre -0.20 e -0.13). A definição do limiar experimental em -0.1203 permitiu isolar com exatidão o tráfego anômalo situado abaixo deste corte.

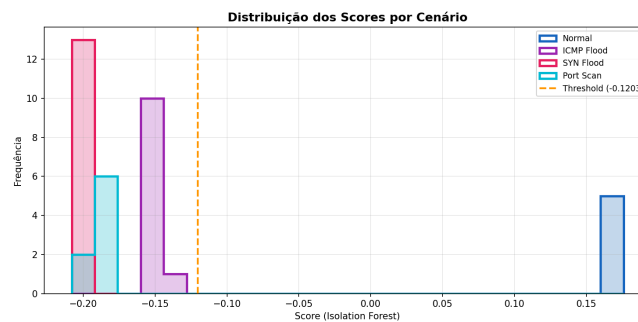


Figura 1. Distribuição do score de anomalia.

4.2 Avaliação Expandida e Desempenho

Neste teste, o *pipeline* operou em tempo real sob requisições simultâneas e varreduras de rede. A Figura 2 ilustra a linha do tempo das janelas de captura e a oscilação do *score* conforme a execução dos ataques em relação ao limiar operacional de -0.1203.

A janela 15 corresponde ao intervalo de transição entre o encerramento do ICMP Flood e o início do SYN Flood, período em que o tráfego retornou brevemente ao padrão benigno, resultando em um score acima do limiar antes da queda imediata provocada pelo ataque subsequente.

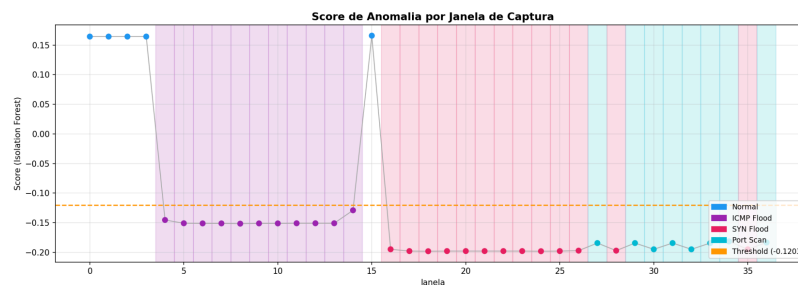


Figura 2. Linha do tempo de detecção de anomalias.

Os ataques de volumetria (ICMP Flood e SYN Flood) provocaram queda imediata no *score* (atingindo *scores* de -0.198 no SYN Flood e -0.151 no ICMP Flood),

mantendo-se continuamente na zona de anomalia abaixo do limiar e assegurando taxa de detecção de 100%.

O *Port Scan* também registrou uma taxa de detecção de 100% (Tabela 1), embora com um comportamento distinto dos ataques de inundação. Ferramentas de varredura, mesmo quando configuradas de forma agressiva, operam com espaçamentos temporais entre as sondagens, gerando picos pontuais de portas únicas de destino. O CyberEye identificou esses desvios em todas as janelas de captura, classificando-os como anômalos abaixo do limiar de -0.1203. A Tabela 1 sintetiza o desempenho global registrado em logs/alerts.csv.

Tabela 1. Taxa de Detecção por Vetor de Ataque.

Cenário	Janelas capturadas	Ataques detectados	Ataques perdidos	Falsos alarmes	Taxa de detecção
ICMP Flood	11	11	0	0	100%
SYN Flood	13	13	0	0	100%
Port Scan	8	8	0	0	100%

5. Considerações Finais

Este trabalho apresentou o CyberEye, um NIDS baseado no algoritmo de ML Não Supervisionado Isolation Forest, desenvolvido como resposta à crescente inadequação dos sistemas de detecção tradicionais frente às ameaças cibernéticas contemporâneas. A proposta foi fundamentada na literatura da área e conduzida seguindo o método Design Science Research (Dresch, Lacerda e Antunes Júnior, 2015), percorrendo as etapas de identificação do problema, seleção do artefato, implementação e avaliação experimental. Os resultados obtidos respondem à questão de pesquisa proposta: modelos baseados em IA, especificamente o Isolation Forest, são capazes de auxiliar na detecção de intrusão em redes ao identificar desvios comportamentais do tráfego sem depender de assinaturas previamente catalogadas.

Os testes conduzidos em ambiente virtualizado demonstraram 100% de taxa de detecção para os três cenários simulados (*ICMP Flood*, *SYN Flood* e *Port Scan*), com zero falsos positivos no tráfego benigno e *threshold* experimental de -0.1203.

Este trabalho apresenta limitações relevantes: os testes foram conduzidos em ambiente virtualizado controlado com cenários de ataque clássicos e volumétricos, o que limita a validade externa dos resultados. O tráfego benigno utilizado era ocioso, tornando a taxa de falsos positivos pouco informativa para cenários reais. O *dataset* CIC-IDS-2017 não cobre tráfego criptografado, restringindo a avaliação a ataques observáveis em texto claro. O overhead de captura em tempo real via Scapy pode ser impeditivo em redes de alta velocidade, demandando soluções mais eficientes.

Como trabalhos futuros, pretende-se: (i) migrar para o uso da biblioteca Libpcap, visando melhorar o desempenho e migrar para análise por fluxos (*flows*); (ii) avaliar o desempenho contra ataques que envolvam criptografia e tunelamento de pacotes; e (iii) testar o sistema em ambientes de rede reais.

Referências

- Alzahrani, A. O. e Alenazi, M. J. F. (2021) "Designing a network intrusion detection system based on machine learning for software defined networks", *Future Internet*, v. 13, n. 5. Disponível em: <https://www.mdpi.com/1999-5903/13/5/111>. Acesso em: 30 mai. 2026.
- Dresch, A., Lacerda, D. P. e Antunes Júnior, J. A. V. (2015) *Design Science Research: Método de Pesquisa para Avanço da Ciência e Tecnologia*. Bookman.
- Liu, F. T., Ting, K. M. e Zhou, Z. (2008) "Isolation Forest", In: IEEE International Conference on Data Mining (ICDM), pp. 413-422.
- Sharafaldin, I., Habibi Lashkari, A. e Ghorbani, A. A. (2018) "Toward Generating a New Intrusion Detection Dataset and Intrusion Traffic Characterization", In: International Conference on Information Systems Security and Privacy (ICISSP), pp. 108-116.
- Fernando, D. W., Komninos, N. e Chen, T. (2020) "A study on the evolution of ransomware detection using machine learning and deep learning techniques", *IoT*. Disponível em: <https://www.mdpi.com/2624-831X/1/2/30>. Acesso em: 22 mai. 2026.
- Ghadermazi, J., Shah, A. e Bastian, N. D. (2024) "Towards real-time network intrusion detection with image-based sequential packets representation", *IEEE Transactions on Big Data*. Disponível em: <https://ieeexplore.ieee.org/document/10535236>. Acesso em: 22 mai. 2026.
- Moamin, S. A. H. et al. (2025) "Artificial Intelligence in Malware and Network Intrusion Detection: A Comprehensive Survey", *Mesopotamian Journal*. Disponível em: <https://mesopotamian.press/journals/index.php/BJAI/article/view/821/752>. Acesso em: 13 mai. 2026.
- Monteiro, C. L. et al. (2026) "Avaliação da eficácia de algoritmos de machine learning na detecção de tráfego malicioso em redes corporativas", *Revista FT*. Disponível em: <https://mail.revistaft.com/ft/article/view/132/124>. Acesso em: 11 mai. 2026.
- Rawindaran, N. et al. (2021) "Cost Benefits of Using Machine Learning Features in NIDS for Cyber Security in UK SME", *Future Internet*. Disponível em: <https://www.mdpi.com/1999-5903/13/8/186>. Acesso em: 22 mai. 2026.
- Shen, M. et al. (2022) "Machine Learning-Powered Encrypted Network Traffic Analysis: A Comprehensive Survey", *IEEE Communications Surveys & Tutorials*. Disponível em: <https://ieeexplore.ieee.org/abstract/document/9896143>. Acesso em: 13 mai. 2026.
- Vanin, P. et al. (2022) "A Study of Network Intrusion Detection Systems Using Artificial Intelligence/Machine Learning", *Applied Sciences*, v. 12, n. 22. Disponível em: <https://www.mdpi.com/2076-3417/12/22/11752>. Acesso em: 13 mai. 2026.