Detecção de Intrusões em Redes sobre a Base Revisada CIC-IDS-2017: Engenharia de Atributos e Avaliação Comparativa entre AdaBoost e XGBoost

Juliana Boudoux Cavalcante Lisbôa[1], **Cleyton Mário de Oliveira Rodrigues**[1]

[1]Escola Politécnica de Pernambuco, Universidade de Pernambuco, Recife, Brasil

jbcl@poli.br, cleyton.rodrigues@upe.br

Abstract. *Network intrusion detection is a relevant challenge for information security. This work develops and compares two supervised classifiers — AdaBoost and XGBoost — for binary classification of network traffic, using a revised version of the CIC-IDS-2017 dataset. The methodology includes feature engineering from IP addresses, transport protocols, and destination ports mapped through the IANA registry. Both classifiers achieved F1-Scores above 99%, with XGBoost obtaining more than an 80% reduction in false positives compared to AdaBoost. Feature importance analysis revealed that engineered attributes, such as Port Cat DNS and Dst IP is Local, rank among the most relevant, validating the proposed feature engineering pipeline.*

Resumo. *A detecção de intrusões em redes é um desafio relevante para a segurança da informação. Este trabalho desenvolve e compara dois classificadores supervisionados — AdaBoost e XGBoost — para classificação binária de tráfego de rede, utilizando uma versão revisada da base CIC-IDS-2017. A metodologia inclui engenharia de atributos a partir de endereços IP, protocolos de transporte e portas de destino mapeadas pelo registro IANA. Ambos os classificadores alcançaram F1-Scores acima de 99%, com o XGBoost obtendo mais de 80% de redução de falsos positivos em comparação ao AdaBoost. A análise de importância de variáveis revelou que atributos criados por engenharia, como Port Cat DNS e Dst IP is Local, figuram entre os mais relevantes, validando o pipeline de engenharia proposto.*

1. Introdução

Com a expansão da Internet e a digitalização dos processos, os dados tornaram-se um dos ativos mais valiosos das organizações. Consequentemente, sua proteção contra acessos não autorizados, fraudes e ataques cibernéticos tornou-se algo relevante. Empresas de todos os tamanhos, governos e instituições investem cada vez mais em soluções de cibersegurança, a fim de preservar as informações que trafegam em suas redes e sistemas.

Diante desse cenário, entende-se a necessidade de buscar soluções inovadoras para os desafios enfrentados no ramo e, para isso, é possível utilizar-se de meios como o Machine Learning (ML). O ML se destaca como uma ferramenta útil no combate às ameaças cibernéticas. A capacidade dessa tecnologia de aprender e se adaptar com base em dados históricos permite a criação de sistemas de detecção de intrusão eficazes. Diferentemente dos métodos tradicionais de detecção, que dependem de assinaturas conhecidas e regras pré-definidas, os modelos baseados em aprendizado de máquina podem

identificar padrões anômalos, identificando até mesmo ataques desconhecidos e zero-day [Soltani et al. 2023].

No entanto, a eficácia de modelos de ML para detecção de intrusão depende fortemente da qualidade dos dados de treinamento. Estudos recentes demonstraram que datasets amplamente utilizados na literatura, como o CIC-IDS-2017 [Sharafaldin et al. 2018], apresentam erros significativos de rotulagem [Liu et al. 2022], o que compromete a validade dos resultados obtidos com esses dados. Além disso, a representação das variáveis de entrada influencia diretamente o desempenho dos classificadores: informações relevantes contidas em campos como endereços IP, protocolos e portas de destino são frequentemente descartadas no pré-processamento, por serem categóricas ou de alta cardinalidade.

Este trabalho aborda essas duas lacunas. Primeiro, utiliza-se a versão revisada da CIC-IDS-2017 proposta por Liu et al. (2022), que inclui correções de rotulagem e novos atributos. Segundo, propõe-se um pipeline de engenharia de atributos que extrai informações relevantes de variáveis tipicamente descartadas. Os classificadores binários avaliados — AdaBoost e XGBoost — foram selecionados a partir de revisão da literatura e posteriormente comparados. Todo o código-fonte desenvolvido para criação de atributos e o treinamento dos modelos foi disponibilizado publicamente no GitHub[1].

A organização deste trabalho está estruturada como descrito a seguir. Na Seção 2, são apresentados os trabalhos relacionados. A Seção 3 detalha a metodologia adotada. Na Seção 4, são discutidos os resultados obtidos. Por fim, a Seção 5 apresenta as considerações finais.

2. Trabalhos Relacionados

Diversos estudos exploram o uso de algoritmos de aprendizado de máquina para detecção de intrusão em redes. Nesta seção são discutidos os trabalhos mais relevantes para o escopo desta pesquisa, organizados segundo o algoritmo utilizado, o dataset adotado e as contribuições metodológicas.

No artigo de Hu et al. (2014), é demonstrado que versões online do AdaBoost podem ser aplicadas em sistemas distribuídos de detecção, possibilitando atualização incremental diante de tráfego dinâmico e atingindo taxas de detecção superiores a 90%, mesmo em cenários com ataques não vistos anteriormente [Hu et al. 2014]. De forma complementar, Zhou et al. (2020) propõem o M-AdaBoost-A, uma extensão do AdaBoost que incorpora a métrica AUC no processo de boosting, alcançando bom desempenho, sobretudo em termos de recall e G-mean [Zhou et al. 2020]. Ainda no contexto do AdaBoost, Yulianto et al. (2019) aplica esse modelo combinado a técnicas de balanceamento e seleção de atributos — SMOTE, PCA e Ensemble Feature Selection (EFS) —, evidenciando o impacto do pré-processamento sobre o desempenho do classificador [Yulianto et al. 2019].

Quanto ao XGBoost, Husain et al. (2019) demonstram que, na base UNSW-NB15, o XGBoost supera algoritmos clássicos como SVM, Naïve Bayes e redes neurais, alcançando acurácia superior a 90% e oferecendo vantagens adicionais em termos de seleção de atributos [Husain et al. 2019]. Jiang et al. (2020) propõem o PSO-XGBoost,

[1] `https://github.com/julianaboudoux/Intrusion-Detection-AdaBoost-XGBoost`

que otimiza os hiperparâmetros do modelo via Particle Swarm Optimization, demonstrando melhorias em precisão, recall e mAP [Jiang et al. 2020].

Em relação aos datasets, a base CIC-IDS-2017 é amplamente utilizada na literatura. No entanto, trabalhos recentes identificaram problemas significativos. Liu et al. (2022) apresenta uma análise detalhada dos erros de rotulagem, geração de atributos e documentação, identificando taxas de corrupção de rótulos de até 95% para determinados tipos de ataque. No mesmo artigo, uma versão revisada do dataset é confeccionada. Essa base resultante é adotada no presente trabalho.

A Tabela 1 sintetiza a comparação entre os trabalhos relacionados e o presente estudo.

Tabela 1. Comparação entre trabalhos relacionados e o presente estudo.

Referência	Algoritmo	Dataset	Criação de Atributos
Hu et al. 2014	AdaBoost (online)	KDD Cup 99	–
Husain et al. 2019	XGBoost	UNSW-NB15	–
Jiang et al. 2020	PSO-XGBoost	NSL-KDD	–
Yulianto et al. 2019	AdaBoost	CIC-IDS-2017	–
Zhou et al. 2020	M-AdaBoost-A	NSL-KDD / AWID	One-hot Encoding
Este trabalho	AdaBoost, XGBoost	CIC-IDS-2017 (melhorada)	Mapeamentos IANA e IP, One-hot Encoding

Diferentemente dos trabalhos anteriores, este estudo combina o uso da base corrigida de Liu et al. (2022) com um pipeline de engenharia de atributos, avaliando comparativamente AdaBoost e XGBoost.

3. Metodologia

A metodologia adotada neste trabalho está organizada nas seguintes etapas: seleção da base de dados, pré-processamento e engenharia de atributos, seleção e implementação dos modelos, e avaliação.

3.1. Seleção da Base de Dados

Foi utilizada a versão aprimorada da CIC-IDS-2017, revisada no estudo de Liu et al. (2022), apresentado na conferência IEEE CNS. Essa pesquisa identificou erros relevantes na rotulagem, geração de features e documentação dos datasets originais, oferecendo uma base de dados corrigida e mais confiável para validação de pesquisas em detecção de intrusão.

O dataset reúne tráfego de rede benigno e malicioso gerado durante cinco dias em uma topologia completa. Após a remoção de valores infinitos, restaram 2.099.971 amostras: 75,93% (1.594.540) benignas e 24,07% (505.431) maliciosas. Embora o tráfego malicioso inclua diversas categorias — como Portscan, DDoS, entre outras — elas foram unificadas no rótulo “Ataque” para atender ao propósito de classificação binária deste estudo. Por fim, os rótulos do tipo “-Attempted” [Liu et al. 2022], que indicam ausência de payload malicioso efetivo, foram re-rotulados como benignos.

3.2. Pré-Processamento e Engenharia de Atributos

O pré-processamento envolveu etapas de limpeza, transformação e codificação de variáveis. Inicialmente, foram removidos entradas contendo valores infinitos ou não numéricos. Em seguida, foram eliminadas variáveis de identificação e não numéricas, como Id, Flow ID, Timestamp, Src IP, Dst IP, Src Port, Dst Port e Protocol.

A fim de preservar informações relevantes dessas variáveis, aplicaram-se técnicas de engenharia de atributos para geração de novas features:

Endereço IP de destino e origem: a partir das colunas Dst IP e Src IP, foram criadas as variáveis binárias Dst IP is Local e Src IP is Local, assumindo valor 1 para endereços privados e 0 para endereços públicos, utilizando a biblioteca ipaddress do Python.

Protocolo: a variável Protocol foi submetida a one-hot encoding, originando as colunas Protocol is 6, Protocol is 17, Protocol is 0 e Protocol is 1, codificadas com valores binários.

Porta de destino (mapeamento IANA): a partir da coluna Dst Port, foi realizado um mapeamento para categorias de serviço de rede com base no registro oficial da IANA (Service Name and Transport Protocol Port Number Registry) [IANA 2026]. Cada porta foi classificada em uma das seguintes categorias: HTTP, DNS, Database, File Transfer, Email, Remote Access, Time Sync e Other. A classificação foi realizada por meio de uma função que analisa o nome do serviço e a descrição associados a cada porta no registro IANA. As categorias resultantes foram codificadas via one-hot encoding, removendo-se a coluna Other para evitar multicolinearidade.

Após a engenharia de atributos, o conjunto de dados foi dividido de forma estratificada (preservando o balanceamento das classes) em treino (70%, 1.469.979 amostras), validação (15%, 314.996 amostras) e teste (15%, 314.996 amostras). Em seguida, aplicou-se o coeficiente de correlação de Pearson com limiar de 0,9 para identificar e remover variáveis altamente correlacionadas, seguindo a abordagem de Liu et al. (2022). Essa etapa resultou em 35 features removidas e um total de 59 features remanescentes.

3.3. Seleção e Implementação dos Modelos

Foram empregados dois modelos de aprendizado supervisionado para a tarefa de classificação binária, selecionados a partir de revisão da literatura: Adaptive Boosting (AdaBoost) e Extreme Gradient Boosting (XGBoost).

O AdaBoost constrói iterativamente um conjunto de classificadores fracos e os combina sequencialmente para formar um classificador robusto [Schapire 1999]. A cada iteração, os pesos das instâncias são ajustados, de modo que exemplos mal classificados recebam maior ênfase no treinamento subsequente.

O XGBoost é uma implementação otimizada de gradient boosting que minimiza diretamente uma função de custo diferenciável, utilizando expansão de Taylor de segunda ordem, regularização L1 e L2, e técnicas avançadas de paralelização [Chen and Guestrin 2016]. A construção de árvores utiliza o método baseado em histogramas (hist-based), que acelera o treinamento em grandes volumes de dados.

Ambos os modelos foram avaliados para diferentes números de estimadores (2,

3, 5, 10, 25 e 40), sendo selecionada a configuração com maior área sob a curva ROC (ROC-AUC) no conjunto de validação. No XGBoost, fixaram-se taxa de aprendizado de 0,1 e profundidade máxima das árvores igual a 3. Para o AdaBoost, utilizou-se o estimador base padrão da biblioteca scikit-learn (árvore de decisão de profundidade 1) com taxa de aprendizado 1,0. Além disso, aplicou-se um ajuste dinâmico do limiar de decisão por meio da maximização do F1-Score, a fim de equilibrar a taxa de detecção de ataques (recall) com a minimização de falsos positivos (precisão). Após a seleção da configuração ótima, cada modelo foi re-treinado com a junção dos conjuntos de treino e validação e avaliado no conjunto de teste.

3.4. Métricas de Avaliação

Na etapa de avaliação foram consideradas métricas de Precisão, Recall, F1-Score e ROC-AUC, além da análise da matriz de confusão. Complementarmente, realizou-se a análise de importância de variáveis por duas abordagens: importância nativa (baseada na contribuição de cada variável para o ganho de informação durante o boosting) e importância por permutação (medindo a queda na ROC-AUC ao embaralhar cada variável no conjunto de teste).

4. Resultados e Discussão

4.1. Desempenho dos Classificadores

A avaliação final de ambos os modelos foi realizada no conjunto de teste, considerando o melhor número de estimadores e o limiar ótimo de decisão selecionados na validação. Os resultados consolidados encontram-se na Tabela 2 e as matrizes de confusão na Tabela 3.

Tabela 2. Resultados da avaliação dos modelos no conjunto de teste.

Modelo	Precisão	Recall	F1-Score	ROC-AUC	Limiar (t^*)
AdaBoost (n=40)	0,9898	0,9928	0,9913	0,9999	0,510
XGBoost (n=40)	0,9980	0,9942	0,9961	0,9998	0,600

Tabela 3. Matriz de confusão (em amostras) no conjunto de teste.

Modelo	Verdadeiro Negativo	Falso Positivo	Falso Negativo	Verdadeiro Positivo
AdaBoost	238.403	779	547	75.267
XGBoost	239.028	154	440	75.374

Ambos os classificadores apresentaram alto desempenho, com F1-Score acima de 0,99. No entanto, a comparação das matrizes de confusão revela diferenças relevantes. O XGBoost produziu 154 falsos positivos, contra 779 do AdaBoost — uma redução de aproximadamente 80%. Essa diferença é particularmente relevante em ambientes de produção, onde cada falso positivo gera um alerta que demanda investigação. Em organizações com alto volume de tráfego, a diferença entre 154 e 779 alertas falsos diários pode representar um impacto significativo na carga operacional.

Em relação aos falsos negativos, ambos os modelos apresentaram valores próximos (547 para o AdaBoost e 440 para o XGBoost). Falsos negativos representam ataques não detectados e, portanto, constituem o cenário mais crítico em um sistema de detecção de intrusão. O recall de ambos os modelos (0,9928 e 0,9942, respectivamente) indica que menos de 1% dos ataques passaram despercebidos.

Cabe ressaltar que a classificação binária adotada neste estudo mascara a dificuldade variável entre tipos de ataque. O dataset contém categorias com volumes muito distintos — Portscan e DoS Hulk representam a maioria do tráfego malicioso, enquanto ataques como Heartbleed (11 amostras), Web Attack - SQL Injection (13) e Web Attack - XSS (18) são extremamente raros. Os falsos negativos reportados podem estar concentrados nessas classes minoritárias, o que uma análise multiclasse permitiria investigar. Essa limitação é discutida na Seção 5.

4.2. Análise de Importância de Variáveis

A Tabela 4 apresenta as dez features mais importantes para cada modelo, utilizando tanto a importância nativa quanto a importância por permutação (ROC-AUC), ordenadas da mais para a menos relevante.

Tabela 4. Top-10 features relevantes por modelo e método de análise.

Rank	AdaBoost (Nativa)	AdaBoost (Permutação)	XGBoost (Nativa)	XGBoost (Permutação)
1	Bwd Packet Length Std	Bwd Packet Length Std	Bwd Packet Length Std	Dst IP is Local
2	Bwd Init Win Bytes	Dst IP is Local	Flow Packets/s	SYN Flag Count
3	Port Cat DNS	Fwd Packet Length Mean	Fwd Packet Length Max	FWD Init Win Bytes
4	Dst IP is Local	FWD Init Win Bytes	SYN Flag Count	Bwd Packet Length Std
5	Fwd Packet Length Mean	Port Cat DNS	Dst IP is Local	Fwd Packet Length Max
6	FWD Init Win Bytes	Flow Packets/s	FWD Init Win Bytes	Total TCP Flow Time
7	Flow Packets/s	Bwd Init Win Bytes	Bwd RST Flags	Flow IAT Min
8	Flow IAT Min	Fwd Packet Length Max	Flow IAT Min	Total Length of Fwd Packet
9	Bwd IAT Max	Bwd PSH Flags	Port Cat File Transfer	ACK Flag Count
10	Bwd PSH Flags	Bwd IAT Total	Port Cat Remote Access	Bwd Packet Length Min

A análise revela que determinados atributos aparecem de forma consistente entre os dois modelos e os dois métodos de avaliação. Bwd Packet Length Std, que mede a variabilidade no tamanho dos pacotes na direção reversa, é o atributo mais importante na análise nativa de ambos os modelos. Essa variabilidade tende a ser discriminativa porque diferentes classes de tráfego apresentam perfis distintos de tamanho de pacote na direção de resposta, característica capturada diretamente por esse atributo.

A feature Dst IP is Local, criada pela engenharia de atributos deste estudo, figura entre as cinco mais importantes em todas as quatro análises, com destaque para a im-

portância por permutação do XGBoost, onde ocupa a primeira posição. Esse resultado indica que a distinção entre tráfego destinado a endereços internos e externos é altamente discriminativa para a identificação de ataques, o que é coerente com a topologia do dataset, na qual o conjunto de máquinas-alvo está concentrado na rede interna.

As features derivadas do mapeamento de portas IANA também demonstraram relevância. Port Cat DNS figura entre as cinco mais importantes tanto na importância nativa quanto por permutação do AdaBoost. A relevância desse atributo pode estar associada a padrões de tráfego na porta 53 gerados por determinados ataques presentes no dataset, hipótese que merece investigação adicional em trabalhos futuros, particularmente por meio de análise multiclasse. No XGBoost, Port Cat File Transfer e Port Cat Remote Access figuram no top-10 da importância nativa, o que é coerente com a composição do dataset, que inclui ataques direcionados a serviços de transferência de arquivos (FTP-Patator) e acesso remoto (SSH-Patator).

A presença dessas features engenheiradas entre as mais relevantes valida o pipeline de engenharia de atributos proposto neste trabalho. Informações que seriam perdidas com a simples remoção das colunas originais (endereços IP, portas e protocolos) foram transformadas em variáveis numéricas que efetivamente contribuem para a discriminação entre tráfego benigno e malicioso.

Outros atributos consistentemente relevantes incluem FWD Init Win Bytes e Bwd Init Win Bytes (tamanho da janela TCP inicial, que varia entre implementações legítimas e ferramentas de ataque), SYN Flag Count (indicador direto de ataques SYN flood e varreduras), e Flow IAT Min (o menor intervalo entre pacotes no fluxo, que tende a ser muito baixo em ataques automatizados).

5. Considerações Finais

Este trabalho apresentou o desenvolvimento e a comparação de dois classificadores baseados em aprendizado de máquina — AdaBoost e XGBoost — para detecção binária de intrusões em redes, utilizando a versão corrigida da base CIC-IDS-2017 e um pipeline de engenharia de atributos que inclui o mapeamento de portas pelo registro IANA.

Os resultados demonstraram alto desempenho em ambos os modelos, com o XGBoost superando o AdaBoost particularmente na redução de falsos positivos (154 contra 779). A análise de importância de variáveis evidenciou que atributos criados pela engenharia proposta — como Dst IP is Local — figuraram entre os mais relevantes, validando a abordagem de transformar informações tipicamente descartadas em features discriminativas.

Algumas limitações devem ser consideradas. A classificação binária agrupa categorias de ataque com volumes e características muito distintas, mascarando a dificuldade variável entre tipos de ataque. Classes raras como Heartbleed (11 amostras) e Web Attack - SQL Injection (13 amostras) provavelmente são sub-representadas nos resultados agregados, e os falsos negativos reportados podem estar concentrados nessas categorias. Além disso, o espaço de hiperparâmetros explorado foi limitado ao número de estimadores, sem variação de parâmetros como learning rate ou profundidade máxima das árvores. Por fim, a base CIC-IDS-2017, mesmo em sua versão corrigida, é composta por tráfego sintético gerado em 2017, o que pode não refletir padrões de ataque contemporâneos.

Como trabalhos futuros, considera-se a extensão para classificação multiclasse, que permitiria investigar o desempenho por tipo de ataque e identificar em quais categorias os modelos apresentam maior dificuldade. Também se propõe a validação em outros datasets, a fim de verificar a robustez dos modelos em diferentes cenários de tráfego.

6. Agradecimentos

Durante a elaboração deste artigo, utilizou-se a ferramenta Claude (modelo Opus 4.6) para auxiliar no aprimoramento e correção textual.

Referências

Chen, T. and Guestrin, C. (2016). XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794.

Hu, W. et al. (2014). Online AdaBoost-based parameterized methods for dynamic distributed network intrusion detection. *IEEE Transactions on Cybernetics*, 44(1):66–82.

Husain, A. et al. (2019). Development of an efficient network intrusion detection model using extreme gradient boosting (XGBoost) on the UNSW-NB15 dataset. In *2019 IEEE International Symposium on Signal Processing and Information Technology (ISSPIT)*, pages 1–7. IEEE.

IANA (2026). Service name and transport protocol port number registry. `https://www.iana.org/assignments/service-names-port-numbers/service-names-port-numbers.xhtml`. Acesso em: 02 mar. 2026.

Jiang, H. et al. (2020). Network intrusion detection based on PSO-XGBoost model. *IEEE Access*, 8:58392–58401.

Liu, L. et al. (2022). Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018. In *2022 IEEE Conference on Communications and Network Security (CNS)*, pages 254–262. IEEE.

Schapire, R. E. (1999). A brief introduction to boosting. In *Proceedings of the 16th International Joint Conference on Artificial Intelligence (IJCAI)*, pages 1401–1406.

Sharafaldin, I. et al. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)*, volume 1, pages 108–116.

Soltani, M. et al. (2023). An adaptable deep learning-based intrusion detection system to zero-day attacks. *Journal of Information Security and Applications*, 76:103516.

Yulianto, A., Sukarno, P., and Suwastika, N. A. (2019). Improving AdaBoost-based intrusion detection system (ids) performance on CIC IDS 2017 dataset. *Journal of Physics: Conference Series*, page 012018.

Zhou, Y., Mazzuchi, T. A., and Sarkani, S. (2020). M-AdaBoost-A based ensemble system for network intrusion detection. *Expert Systems with Applications*, 162:113864.