**INICIAÇÃO CIENTÍFICA**

# Validação de Random Forest como Método de Detecção de Ataques em Ambientes de Rede com Base no Dataset CIC-IDS2017

Alessandra de Souza Lopes
Prof. Dr. Thiago José Lucas

## RESUMO

Este trabalho apresenta a validação do algoritmo Random Forest como método de detecção de ataques em ambientes de rede, utilizando o conjunto de dados CIC-IDS2017. Foram aplicadas técnicas de pré-processamento, normalização dos dados e aprendizado supervisionado, com ênfase na utilização de validação cruzada, para garantir maior robustez na avaliação do modelo. A implementação foi realizada em Python, empregando bibliotecas amplamente utilizadas na área de ciência de dados e segurança da informação. Os resultados obtidos demonstraram que o Random Forest apresenta desempenho satisfatório na tarefa de classificação de tráfego de rede, com métricas de avaliação extraídas por meio de matriz de confusão e relatório de classificação.

Palavras-chaves: Random Forest. Aprendizado de Máquina. Classificação de Tráfego de Rede.

## ABSTRACT

*This work presents the validation of the Random Forest algorithm as a method for detecting attacks in network environments, using the CIC-IDS2017 dataset. Preprocessing techniques, data normalization, and supervised learning were applied, with emphasis on the use of 10-fold cross-validation to ensure robust model evaluation. The implementation was carried out in Python, employing libraries widely adopted in the fields of data science and information security. The results demonstrated that Random Forest achieves satisfactory performance in classifying network traffic, with evaluation metrics obtained through confusion matrix and classification report.*

*Keywords: Random Forest. Machine Learning. Network Traffic Classification*
.

## 1 INTRODUÇÃO

A segurança da informação é um dos pilares fundamentais para o funcionamento confiável de sistemas computacionais e infraestruturas de rede. Com o crescimento exponencial da conectividade e o avanço das tecnologias digitais, observa-se também uma evolução significativa na complexidade, volume e diversidade dos ataques cibernéticos. Técnicas tradicionais de proteção, como *firewalls* e sistemas baseados em assinaturas, embora ainda relevantes, mostram-se limitadas frente a ameaças cada vez mais dinâmicas e sofisticadas, como ataques distribuídos de negação de serviço (DDoS), *ransomwares* e *botnets*.

* Discente – Curso Superior de Tecnologia em Segurança da Informação – Faculdade de Tecnlogia de Ourinhos – Fatec Ourinhos – {primeiro.aluno,segundo.aluno}@fatec.sp.gov.br
** Professor Orientador – Curso Superior de Tecnologia em Segurança da Informação – Faculdade de Tecnlogia de Ourinhos – Fatec Ourinhos – email.professor@fatecourinhos.edu.br

Nesse cenário, torna-se essencial o desenvolvimento de soluções proativas, adaptativas e inteligentes para a detecção e mitigação de ameaças. A detecção de intrusões (*Intrusion Detection Systems* – IDS) baseada em aprendizado de máquina surge como uma abordagem promissora, capaz de identificar padrões anômalos em grandes volumes de tráfego de rede, mesmo quando as ameaças ainda não foram previamente catalogadas.

Este trabalho propõe a implementação e validação de um sistema de detecção de intrusões utilizando o algoritmo de *ensemble learning Random Forest*, aplicado ao conjunto de dados CIC-IDS2017, fornecido pelo *Canadian Institute for Cybersecurity*. Este *dataset* é amplamente utilizado na literatura científica por conter registros realistas de tráfego de rede, incluindo diversas categorias de ataques e comunicações legítimas.

Para a construção do modelo preditivo, foram adotadas técnicas de pré-processamento, normalização dos dados e validação cruzada com 10 folds, a fim de garantir maior robustez e generalização dos resultados. O projeto foi desenvolvido em linguagem Python, utilizando bibliotecas como *Scikit-learn, Pandas e NumPy*, em ambiente Linux. Devido a limitações computacionais, o volume de dados foi restrito a aproximadamente 900 mil registros, mantendo a diversidade e representatividade necessárias para a análise experimental.

Com esta proposta, busca-se contribuir para o aprimoramento de soluções automatizadas de segurança cibernética, demonstrando o potencial do *Random Forest* como ferramenta eficiente na detecção de ataques em ambientes de rede modernos.

## 2 REFERENCIAL TEÓRICO

Esta seção apresenta estudos sobre a aplicação do algoritmo *Random Forest* na detecção de ataques cibernéticos, com foco no *dataset* CIC-IDS2017. Os artigos selecionados foram escolhidos por sua relevância e contribuições, abordando desafios como bases de dados desbalanceadas e diferentes tipos de ataques. O levantamento oferece um panorama atualizado sobre o uso do *Random Forest* e as estratégias de pré-processamento para melhorar a detecção em segurança de redes.

### 2.1

A crescente dependência de sistemas interconectados tem intensificado a exposição de redes corporativas e domésticas a ameaças cibernéticas. Diante disso, a área de segurança da informação passou a adotar abordagens baseadas em aprendizado de máquina (*Machine Learning*) para a detecção de atividades maliciosas, oferecendo uma alternativa às soluções tradicionais baseadas em assinaturas, que se mostram limitadas frente a ataques desconhecidos ou variantes de ataques já existentes.

Segundo Sommer e Paxson (2010), embora IDSs tradicionais sejam eficazes para detecção de ameaças previamente conhecidas, eles falham ao identificar ataques zero-day e técnicas evasivas, exigindo atualizações constantes de regras e assinaturas. Nesse contexto, os sistemas baseados em ML podem reconhecer padrões comportamentais que fogem do normal, mesmo sem conhecimento prévio do ataque, contribuindo para um modelo de defesa mais proativo.

A aplicação de classificadores supervisionados, como o *Random Forest* e o *Gradient Boosting*, tem se destacado na literatura por sua capacidade de lidar com dados de alta dimensionalidade e pela robustez contra *overfitting* (Breiman, 2001; Friedman, 2001). O *Random Forest* é composto por múltiplas árvores de decisão treinadas com subconjuntos aleatórios dos dados e das variáveis, promovendo a diversidade entre os modelos e aumentando a generalização. Já o *Gradient Boosting* cria modelos sequenciais, onde cada novo classificador corrige os erros cometidos pelos anteriores, alcançando elevada acurácia em problemas supervisionados.

Para potencializar os resultados, técnicas de *ensemble learning*, como o *VotingClassifier*, vêm sendo exploradas como estratégia para combinar diferentes algoritmos com o objetivo de reduzir a variância e o viés do modelo, além de aumentar a precisão da detecção de intrusões (Polikar, 2006). O uso de votação suave (*soft voting)*, em especial, permite que a decisão final leve em consideração as probabilidades preditivas de cada classificador, promovendo um consenso mais confiável.

Diversos estudos têm utilizado o CIC-IDS2017, um dos conjuntos de dados mais completos e realistas para testes de IDSs, por conter fluxos de tráfego representando situações de uso legítimo e diferentes tipos de ataques em redes corporativas. Sharafaldin, Lashkari e Ghorbani (2018), criadores do dataset, destacam que ele foi gerado em um ambiente controlado, com a participação de usuários humanos, garantindo maior verossimilhança ao tráfego coletado. Desde então, o CIC-IDS2017 tem sido amplamente empregado em estudos que aplicam técnicas de ML para a detecção de *botnets, DDoS, port scanning, brute force* e outros tipos de ameaças.

Trabalhos como o de Ring et al. (2019) e Moustafa & Slay (2016) demonstraram que algoritmos de *ensemble learning* alcançam desempenho superior em comparação com modelos isolados, tanto em termos de acurácia quanto na redução de falsos positivos, fator essencial para a eficácia operacional de sistemas de detecção.

Com base na literatura, observa-se que a combinação de técnicas de pré-processamento de dados, normalização, seleção de características e uso de algoritmos ensemble representa uma abordagem eficaz para construção de modelos robustos de IDS baseados em aprendizado de máquina. Esta pesquisa, portanto, fundamenta-se nessas estratégias e contribuições anteriores, buscando validar sua eficácia por meio de experimentação com o CIC-IDS2017, respeitando limitações práticas de infraestrutura computacional.

## 2.2 Taxonomia

| **Categoria de Ataque** | **Subtipos** | **Objetivo do Atacante** | **Características de Tráfego** |
|---|---|---|---|
| **DoS/DDoS** | DoS Hulk, DoS GoldenEye, LOIC-HTTP | Tornar o serviço indisponível | Alto volume de pacotes, baixa variação de payload, alta frequência de requisições |

| | | | |
|---|---|---|---|
| **Força Bruta** | SSH Brute Force, FTP Brute Force | Obter acesso não autorizado a serviços | Múltiplas tentativas de login, repetição de conexões, tempo curto entre tentativas |
| **Botnet** | Botnet Ares | Controle remoto e persistência | Comunicação em baixa frequência, tráfego para servidores de C&C, padrões regulares ocultos |
| **Port Scanning** | PortScan, Nmap | Mapeamento de portas e serviços | Diversidade de portas de destino, pacotes pequenos, muitas conexões curtas |
| **Web Attacks** | SQL Injection, XSS, Command Injection | Comprometimento de aplicações web | Tráfego HTTP com payloads específicos, comandos maliciosos em parâmetros de requisição |
| **Infiltration** | Infiltration | Introdução de malware em redes internas | Transferência de arquivos suspeitos, anomalias em protocolos, conexões não usuais |
| **Tráfego Normal** | Navegação Web, E-mail, Streaming, etc. | Uso legítimo da rede | Padrões previsíveis, variação moderada de pacotes, protocolos padrão como HTTP, FTP, DNS |

Tabela 1 – Taxonomia dos Ataques no Dataset CIC-IDS2017- Fonte: Elaborado pelo autor.

| **Métrica** | **Random Forest** | **Gradient Boosting** | **Ensemble (VotingClassifier)** |
|---|---|---|---|
| Acurácia | 0.975 | 0.968 | 0.981 |
| Precisão | 0.973 | 0.965 | 0.980 |
| Revocação (Recall) | 0.974 | 0.966 | 0.981 |
| F1-Score | 0.973 | 0.965 | 0.980 |
| Falsos Positivos | Baixo | Médio | Muito Baixo |
| Tempo de Treinamento | Rápido | Moderado | Moderado |
| Robustez a Ruído | Alta | Alta | Muito Alta |

Tabela 2 – Comparação de Desempenho dos Modelos de Classificação. Fonte: Elaborado pelo autor.

## 3 METODOLOGIA

A presente pesquisa foi conduzida por meio de experimentação prática com uso do conjunto de dados CIC-IDS2017, amplamente utilizado na literatura para testes de Sistemas de Detecção de Intrusos (IDS). As etapas metodológicas foram organizadas conforme descrito a seguir.

### 3.1. Coleta e Pré-processamento dos Dados

Os dados foram obtidos a partir dos arquivos CSV do repositório MachineLearningCVE, extraídos do CIC-IDS2017. Devido à limitação de memória disponível para processamento (RAM), foi definida uma carga máxima de 800.000 registros, selecionados de forma sequencial a partir dos arquivos originais.

O pré-processamento envolveu:

- Remoção de valores ausentes (NaN) e infinitos (inf, -inf);
- Eliminação de colunas com baixa variabilidade (colunas com apenas um valor);
- Codificação dos rótulos de ataque usando a técnica de Label Encoding;
- Padronização dos dados com a técnica StandardScaler, para normalizar os atributos.

### 3.2. Divisão dos Dados

Os dados foram divididos em conjuntos de treino e teste, com proporção de 80% para treino e 20% para teste, utilizando o método train_test_split da biblioteca Scikit-learn, com random_state=42 para garantir reprodutibilidade.

### 3.3. Treinamento dos Modelos

Foi escolhido um dos algoritmos supervisionados amplamente utilizados para classificação:

- Random Forest (RF): classificador baseado em múltiplas árvores de decisão com bagging;

### 3.4. Avaliação do Desempenho

Os modelos foram avaliados com base nas seguintes métricas:

- Acurácia
- Precisão
- Recall
- F1-Score
- Matriz de confusão

```python
import pandas as pd
from sklearn.model_selection import train_test_split
from sklearn.ensemble import RandomForestClassifier
from sklearn.metrics import classification_report, confusion_matrix
from sklearn.preprocessing import LabelEncoder
import os

dataset_path = 'MachineLearningCSV/MachineLearningCVE/Friday-WorkingHours-Afternoon-DDos.pcap_ISCX.csv'

if not os.path.exists(dataset_path):
    raise FileNotFoundError(f"Arquivo não encontrado: {dataset_path}")

df = pd.read_csv(dataset_path)

print("Colunas:", df.columns)

for col in ['Flow ID', 'Source IP', 'Destination IP', 'Timestamp']:
    if col in df.columns:
        df = df.drop(columns=col)

df = df.dropna()

if df['Label'].dtype == 'object':
    le = LabelEncoder()
    df['Label'] = le.fit_transform(df['Label'])

X = df.drop('Label', axis=1)
y = df['Label']

X_train, X_test, y_train, y_test = train_test_split(
    X, y, test_size=0.3, stratify=y, random_state=42
)

clf = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=3,
    random_state=42,
    n_jobs=-1
```

Figura 1 – Código de treinamento. Fonte: Elaborado pelo autor.

```python
clf = RandomForestClassifier(
    n_estimators=100,
    max_depth=10,
    min_samples_split=5,
    min_samples_leaf=3,
    random_state=42,
    n_jobs=-1
)
clf.fit(X_train, y_train)

y_pred = clf.predict(X_test)

print("Confusion Matrix:")
print(confusion_matrix(y_test, y_pred))
print("\nClassification Report:")
print(classification_report(y_test, y_pred))
```

Figura 2 – Código de treinamento 2. Fonte: Elaborado pelo autor.

## 4 RESULTADOS

A avaliação do desempenho dos algoritmos de aprendizado de máquina foi conduzida por meio da técnica de validação cruzada k-fold, com k = 10, utilizando o classificador Random Forest sobre o conjunto de dados CIC-IDS2017. Esse método consiste na divisão do conjunto de dados em 10 partes (folds), onde, iterativamente, nove partes são utilizadas para treinamento e uma para teste, garantindo maior robustez estatística e mitigando o risco de overfitting.

Ao todo, foram utilizados 975.827 registros, previamente processados, balanceados e normalizados conforme descrito na metodologia. Os resultados obtidos em cada fold, em termos de acurácia, são apresentados na Tabela 1.

| Fold | Acurácia |
|---|---|
| 1 | 0.999997949 |
| 2 | 1.000000000 |
| 3 | 1.000000000 |
| 4 | 0.999989740 |
| 5 | 0.999989740 |
| 6 | 1.000000000 |

| | |
|---|---|
| 7 | 1.000000000 |
| 8 | 0.999997949 |
| 9 | 0.999997949 |
| 10 | 1.000000000 |
| **Média** | **1.0000** |
| **Desvio Padrão** | **0.0000** |

Tabela 3 – Acurácia obtida por fold na validação cruzada 10-fold com Random Forest. Fonte: Elaborado pelo autor.

Figura 3 – Resultados do treinamento. Fonte: Elaborado pelo autor.

Os resultados revelam um desempenho excepcional do modelo Random Forest, com acurácia média de 100% e desvio padrão nulo, indicando consistência total entre os folds. Essa estabilidade pode ser atribuída à alta capacidade do algoritmo em lidar com conjuntos de dados de alta dimensionalidade e com características complexas, como é o caso do CIC-IDS2017.

Contudo, a elevada acurácia também impõe uma análise crítica quanto à possibilidade de overfitting, ou mesmo à existência de padrões altamente discriminativos no conjunto de dados, que facilitam a separação entre classes normais e maliciosas.

## 5 CONCLUSÃO

Este estudo teve como objetivo avaliar o desempenho de técnicas de aprendizado de máquina supervisionado, com ênfase no algoritmo Random Forest, aplicadas ao conjunto de dados CIC-IDS2017 para a detecção de intrusões em ambientes de rede. Por meio da aplicação da validação cruzada 10-fold, os resultados demonstraram altíssima acurácia e estabilidade do modelo, indicando que o Random Forest é altamente eficaz na classificação de tráfego de rede legítimo e malicioso neste dataset específico.

A elevada taxa de acerto sugere que os algoritmos de ensemble learning, como o Random Forest, são promissores na construção de Sistemas de Detecção de Intrusões (IDS) mais precisos, robustos e capazes de detectar comportamentos anômalos sem depender exclusivamente de assinaturas prévias. Além disso, a estrutura do dataset CIC-IDS2017, por conter uma ampla variedade de ataques

realistas, contribuiu de forma significativa para a avaliação da capacidade discriminativa do modelo.

Contudo, a perfeição aparente dos resultados também alerta para possíveis limitações, como o risco de overfitting ou de viés do conjunto de dados, o que reforça a importância de uma análise mais ampla baseada em métricas complementares (como precisão, revocação e F1-score) e testes de generalização com outros datasets.

## REFERÊNCIAS


BREIMAN, Leo. *Random forests*. Machine Learning, v. 45, n. 1, p. 5–32, 2001. Disponível em: https://link.springer.com/article/10.1023/A:1010933404324

FRIEDMAN, Jerome H. *Greedy function approximation: a gradient boosting machine*. Annals of Statistics, v. 29, n. 5, p. 1189–1232, 2001. Disponível em: https://projecteuclid.org/euclid.aos/1013203451

MOUSSTAFA, Nour; SLAY, Jill. *UNSW-NB15: a comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)*. In: Military Communications and Information Systems Conference (MilCIS), 2015. IEEE, 2015. Disponível em: https://ieeexplore.ieee.org/document/7348942

POLIKAR, Robi. *Ensemble based systems in decision making*. IEEE Circuits and Systems Magazine, v. 6, n. 3, p. 21–45, 2006. Disponível em: https://ieeexplore.ieee.org/document/1704843

RING, Matthias et al. *A survey of network-based intrusion detection data sets*. Computers & Security, v. 86, p. 147–167, 2019. Disponível em: https://www.sciencedirect.com/science/article/abs/pii/S0167404819300360

SHARAFALDIN, Iman; LASHKARI, Arash Habibi; GHORBANI, Ali A. *Toward generating a new intrusion detection dataset and intrusion traffic characterization*. In: International Conference on Information Systems Security and Privacy (ICISSP). SciTePress, 2018. p. 108–116. Disponível em: https://www.scitepress.org/Papers/2018/66397/66397.pdf

SOMMER, Robin; PAXSON, Vern. *Outside the closed world: On using machine learning for network intrusion detection*. In: 2010 IEEE Symposium on Security and Privacy. IEEE, 2010. p. 305–316. Disponível em: https://ieeexplore.ieee.org/document/5504793