Classificação de Ataques de Rede com Random Forest e SVM

Utilizando o Dataset CIC-IDS2017

Jhonatas Gomes Ribeiro

Instituto Federal de Ciência e Tecnologia do Piauí - IFPI

Introdução e Contexto

Problema

A crescente sofisticação das ameaças cibernéticas e a proliferação de dispositivos conectados tornam a segurança de redes uma preocupação central.

Solução

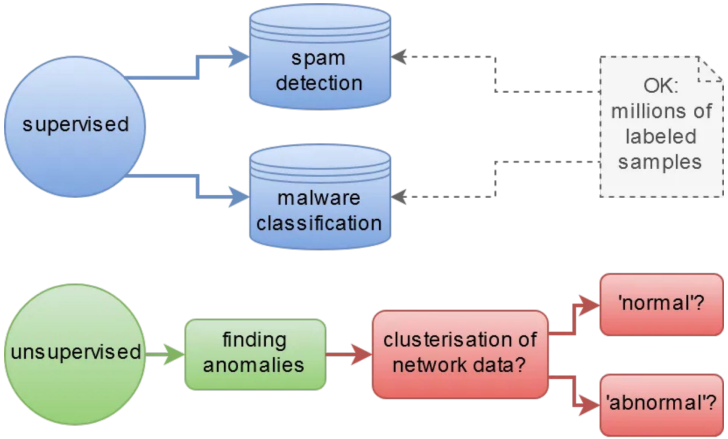
Sistemas de Detecção de Intrusões (IDS) eficazes são essenciais para identificar atividades maliciosas, garantindo a integridade e segurança da rede.

Nosso Desafio

Explorar o potencial do Aprendizado de Máquina para aprimorar a capacidade dos IDS em reconhecer padrões de ataque.

Objetivo do Trabalho

Classificar diferentes tipos de ataques de rede usando Random Forest e SVM no dataset CIC-IDS2017, avaliando e comparando o desempenho de cada modelo.



O Dataset: CIC-IDS2017

Visão Geral

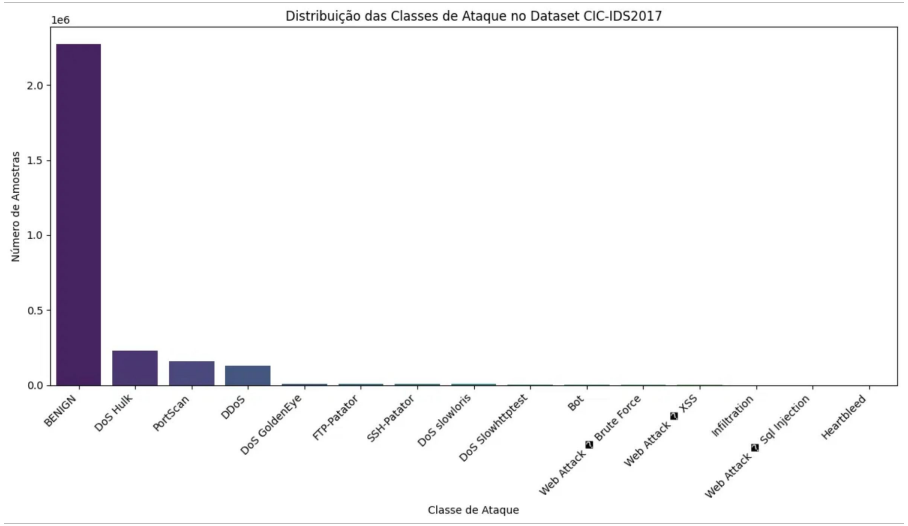
Dataset de detecção de intrusões reconhecido por ser completo e realista.

Metodologia de Coleta

Construído com base em 11 critérios essenciais para um benchmark confiável, simulando tráfego real com o sistema B-Profile.

Dados

- Período: 3 a 7 de julho de 2017 (5 dias úteis, segunda-feira apenas tráfego normal)
- Ataques Incluídos: Brute Force (FTP, SSH), DoS, Heartbleed, Web Attack (Brute Force, XSS, SQL Injection), Infiltration, Botnet e DDoS



Distribuição das Classes de Ataque no Dataset CIC-IDS2017

Observação:

Notável desbalanceamento de classes, com a maioria das amostras sendo benignas.

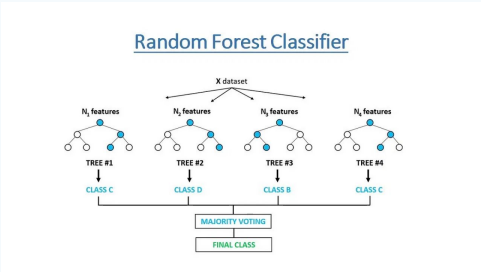
Algoritmos de Classificação

Random Forest

Algoritmo baseado em ensembles que constrói múltiplas árvores de decisão.

Combina previsões de árvores individuais para determinar a classe final, resultando em um modelo robusto.

Permite analisar a importância das características para entender quais atributos são mais relevantes na detecção de ataques.

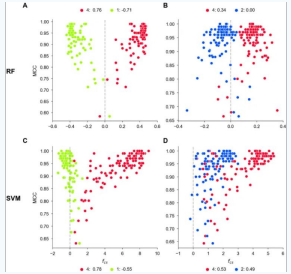


Support Vector Machine (SVM)

Algoritmo de aprendizado de máquina que busca um hiperplano para separar classes em um espaço de alta dimensionalidade.

Eficaz em dados de rede com alta dimensionalidade e classes balanceadas.

LinearSVC: Implementação mais rápida do SVC para o caso linear, recomendada para grandes datasets e flexível para dados não linearmente separáveis.



Técnicas Essenciais

Pré-Processamento de Dados

Limpeza (valores infinitos/ausentes), padronização de nomes de colunas e remoção de duplicatas para garantir a qualidade dos dados.

Seleção de Características

Reduz a dimensionalidade dos dados e melhora o desempenho do modelo, identificando os atributos mais relevantes.

SMOTE

Cria amostras sintéticas para classes minoritárias, balanceando o conjunto de dados de treinamento e melhorando a capacidade do modelo de identificar ataques.

Metodologia: Etapas do Trabalho

1 Carregamento e Visão dos Dados

Concatenação de 8 arquivos CSV, resultando em 2.8M linhas e 79 colunas.

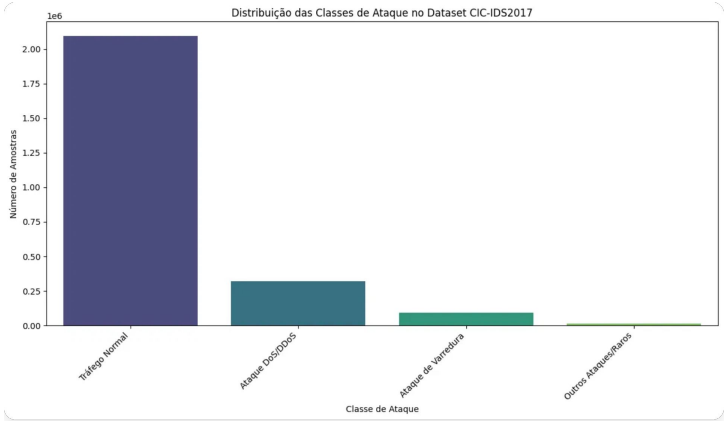
2 Análise Exploratória

Identificação do desbalanceamento de classes (ver gráfico).

3 Pré-Processamento

Padronização de colunas, tratamento de NaNs/infinitos, remoção de duplicatas.

Limpeza e agrupamento de rótulos (Ex: DoS Hulk, DDoS → Ataque DoS/DDoS).



Distribuição das Classes de Ataque após Agrupamento

Observação:
Após o agrupamento, o desbalanceamento ainda persiste, mas com menos classes para classificar.

Metodologia: Preparação Final dos Dados

Divisão e Escalonamento

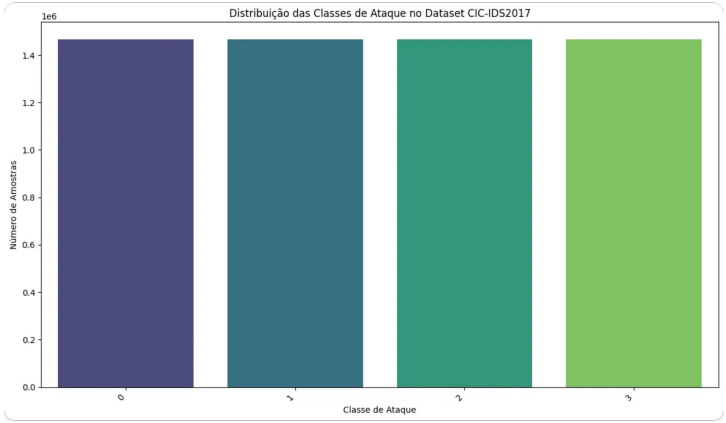
1. Separação em X (features) e y (rótulos)
2. Codificação de rótulos (texto para numérico)
3. Remoção de colunas não numéricas e redução da precisão dos dados
4. Divisão treino/teste (70/30) com estratificação
5. Escalonamento das features com StandardScaler
6. Remoção de colunas sem variância

Seleção de Características

1. Uso de SelectKBest para selecionar as 30 características mais relevantes

Balanceamento com SMOTE

1. Aplicação do SMOTE para equalizar o número de amostras em todas as classes



Distribuição das Classes de Ataque após SMOTE

Observação:

Após a aplicação do SMOTE, todas as classes possuem o mesmo número de amostras, eliminando o desbalanceamento.

Treinamento e Avaliação dos Modelos

Random Forest

Instanciação:
RandomForestClassifier com random_state=42 e n_jobs=-1 para utilizar todos os núcleos da CPU

Análise de Importância:
Utilização de feature_importances_ para identificar as características mais relevantes

SVM (LinearSVC)

Instanciação:
LinearSVC com random_state=42 e max_iter=10000 para garantir convergência

Análise de Importância:
Utilização dos coeficientes do modelo (coef_) para inferir a importância das características

Etapas Comuns

Treinamento
Ajuste do modelo aos dados de treino balanceados e com características selecionadas

Inferência
Realização de previsões no conjunto de teste e medição do tempo de inferência

Avaliação
Cálculo de métricas: acurácia, precisão, recall, F1-score e matriz de confusão

Comparação
Análise comparativa entre os modelos baseada em métricas de desempenho e tempos de execução

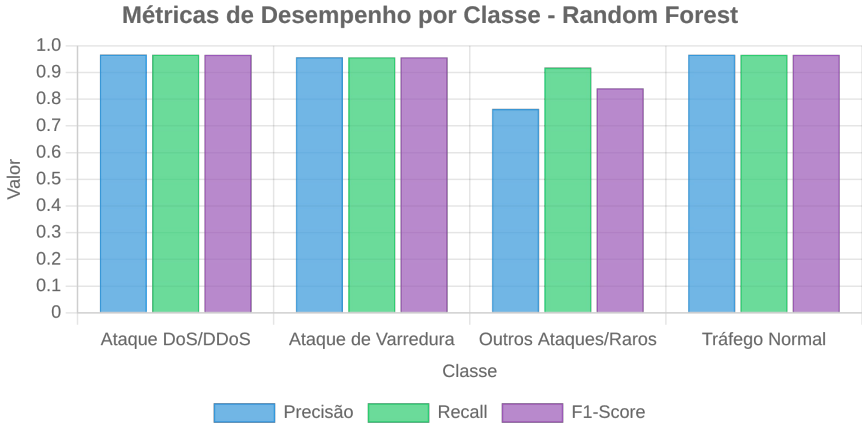
Resultados: Random Forest

Métricas de Classificação

Acurácia	F1-Score (ponderado)
0.9968	**0.9968**

Precisão (ponderada)	Recall (ponderado)
0.9970	**0.9968**

Visualização de Desempenho



Desempenho por Classe

Classe de Ataque	Precisão	Recall	F1-Score
Ataque DoS/DDoS	1.00	1.00	1.00
Ataque de Varredura	0.99	0.99	0.99
Outros Ataques/Raros	0.79	0.95	0.87
Tráfego Normal	1.00	1.00	1.00

Eficiência Computacional

Tempo de Treinamento	Tempo de Inferência
10.50 minutos	**2.32 segundos**

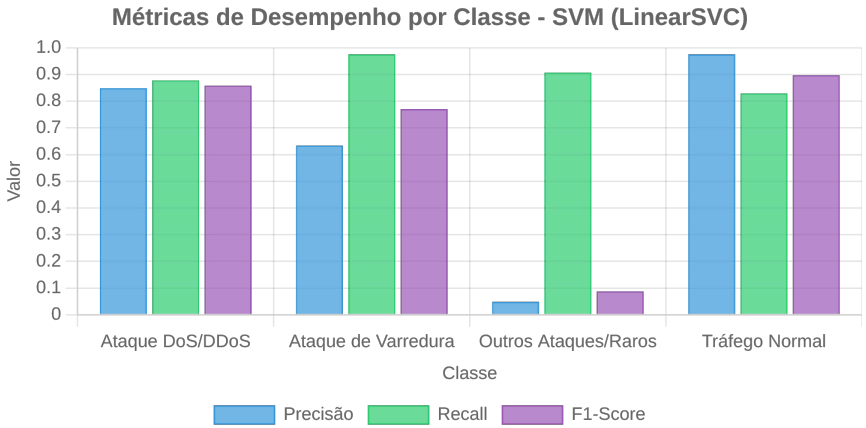
Resultados: SVM (LinearSVC)

Métricas de Classificação

Acurácia	F1-Score (ponderado)
0.8615	**0.9044**

Precisão (ponderada)	Recall (ponderado)
0.9644	**0.8615**

Visualização de Desempenho



Desempenho por Classe

Classe de Ataque	Precisão	Recall	F1-Score
Ataque DoS/DDoS	0.87	0.90	0.88
Ataque de Varredura	0.65	1.00	0.79
Outros Ataques/Raros	0.05	0.93	0.09
Tráfego Normal	1.00	0.85	0.92

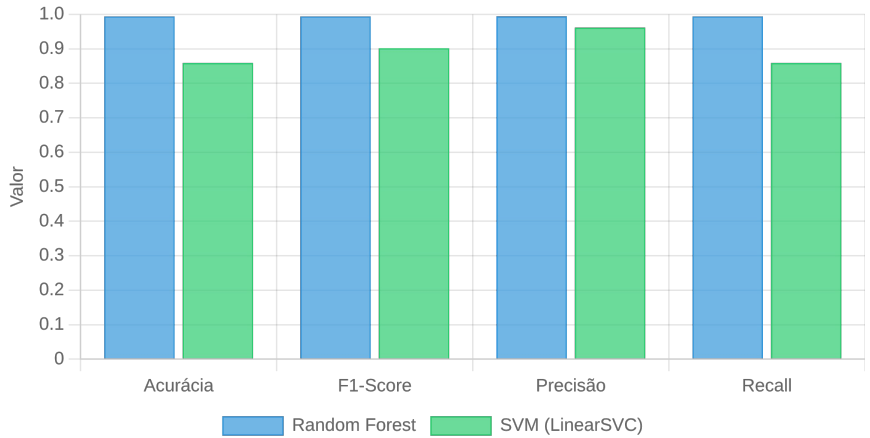
Eficiência Computacional

Tempo de Treinamento	Tempo de Inferência
23.84 minutos	**0.23 segundos**

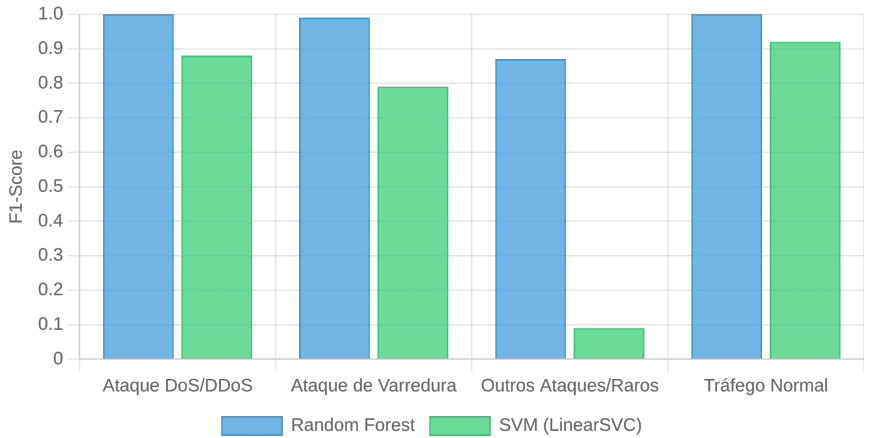
Discussão Comparativa

Desempenho de Classificação

Métricas Gerais



F1-Score por Classe



Análise de Desempenho

Random Forest: Superou o SVM em todas as métricas, alcançando acurácia próxima de 100%. Seu desempenho em "Outros Ataques/Raros" (F1: 0.87) demonstrou a eficácia do SMOTE.

SVM: Acurácia geral inferior (0.86). Desempenho limitado com classes minoritárias (F1: 0.09 para "Outros Ataques/Raros") e ataques específicos.

Eficiência Computacional

Treinamento: Random Forest foi mais rápido (10.50 min vs 23.84 min do SVM).

Inferência: SVM foi significativamente mais rápido (0.23 seg vs 2.32 seg do Random Forest).

Trade-off: Agilidade do SVM na inferência é crítica para detecção em tempo real, mas Random Forest oferece maior precisão.

Conclusão e Trabalhos Futuros

Conclusão Principal

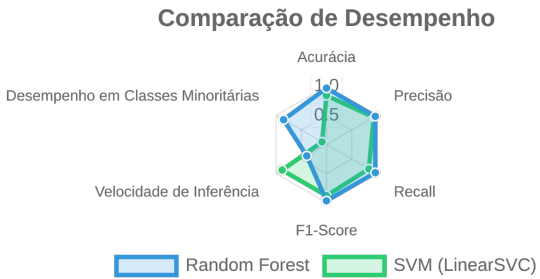
Random Forest foi a melhor escolha para a classificação de ataques no CIC-IDS2017, com acurácia geral de 0.99. Sua capacidade de generalização, impulsionada pelo SMOTE e seleção de características, foi evidente.

Limitações do SVM

Desempenho inferior em classes minoritárias, apesar da inferência rápida. Precisão baixa (0.05) para "Outros Ataques/Raros" indica dificuldade em generalizar para classes menos representadas.

Importância das Etapas

O trabalho ressalta a importância de pré-processamento, seleção de características e balanceamento de classes para IDSs eficazes. Estas etapas foram cruciais para o bom desempenho dos modelos.



Trabalhos Futuros

- Explorar outras estratégias de kernel em SVM para melhorar o desempenho em classes minoritárias.
- Investigar outras técnicas de seleção de características para identificar atributos mais relevantes para cada tipo de ataque.
- Aplicar modelos de deep learning para desempenho aprimorado, especialmente em cenários com grande volume de dados.
- Desenvolver sistemas híbridos que combinem a precisão do Random Forest com a velocidade de inferência do SVM.
- Implementar e testar os modelos em ambientes de rede reais para validar sua eficácia em condições operacionais.

Perguntas?

Obrigado pela atenção!

Jhonatas Gomes Ribeiro

Instituto Federal de Ciência e Tecnologia do Piauí - IFPI

jhonatasgomes2003@gmail.com