# 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 no cenário tecnológico mundial.

## Solução

Sistemas de Detecção de Intrusões (IDS) eficazes são essenciais para identificar atividades maliciosas na rede, garantindo integridade, confidencialidade e disponibilidade das informações.

## Desafio

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

## 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 em termos de métricas como acurácia, precisão, recall e F1-score.

> *"A utilização de técnicas de aprendizado de máquina surge como desafio importante para aprimorar a capacidade dos IDS's em reconhecer padrões de ataque de rede."*

# O Dataset: CIC-IDS2017

## Visão Geral

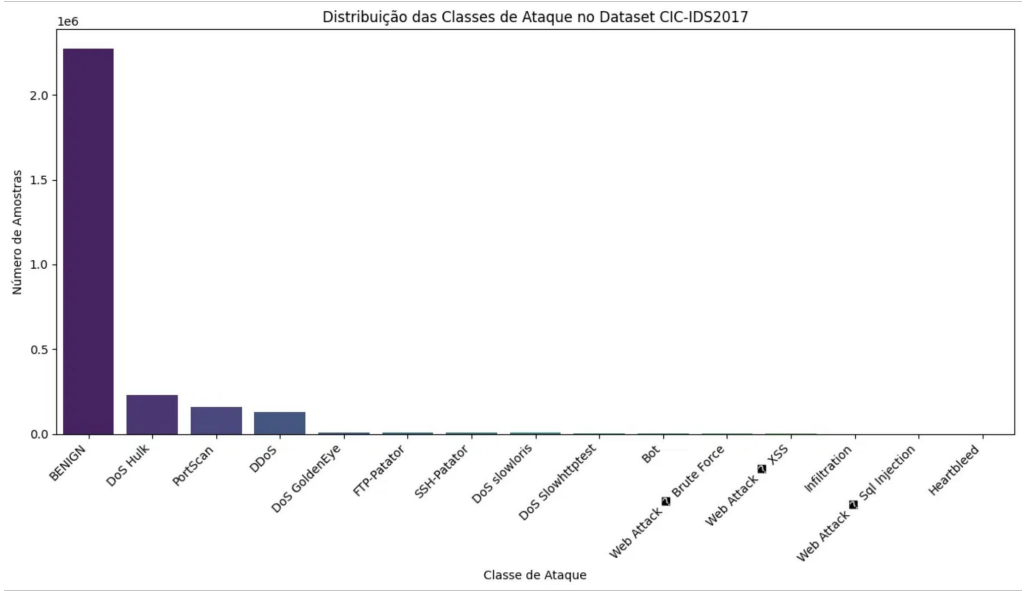
O CIC-IDS2017 (Canadian Institute for Cybersecurity - Intrusion Detection System 2017) é um dataset de detecção de intrusões reconhecido por ser completo e realista, simulando cenários de rede próximos aos ambientes reais.

## Coleta de Dados

A coleta ocorreu de 3 a 7 de julho de 2017 (5 dias úteis), com tráfego de segunda-feira sendo apenas atividades normais. O sistema B-Profile foi utilizado para simular comportamento humano e gerar tráfego baseado em protocolos de rede.

## Tipos de Ataques

- Brute Force (FTP e SSH)
- DoS e DDoS (Hulk, GoldenEye, Slowloris, Slowhttptest)
- Web Attack (Brute Force, XSS, SQL Injection)
- Infiltration, Botnet, Heartbleed, Port Scan




*Distribuição das Classes de Ataque no Dataset CIC-IDS2017 - Notável desbalanceamento com predominância de tráfego normal*

**Composição do Dataset**

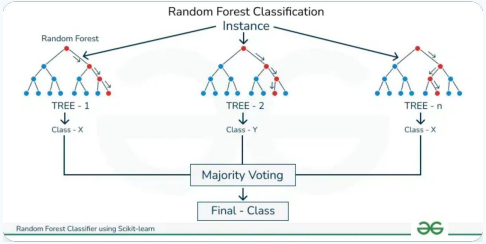
8 arquivos CSV, resultando em 2.830.743 linhas e 79 colunas após concatenação.

# Algoritmos de Classificação

## Random Forest

Algoritmo de ensemble que utiliza múltiplas árvores de decisão para classificação, combinando seus resultados através de votação majoritária.

- **Funcionamento:**
  Cria múltiplas árvores usando subconjuntos aleatórios de dados.
- **Bagging:**
  Amostragem com reposição para diferentes conjuntos de treino.




### Vantagens

- Alta precisão
- Robusto a overfitting
- Lida bem com dados desbalanceados

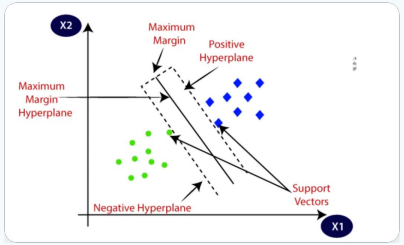
### Desvantagens

- Computacionalmente intensivo
- Modelos grandes (memória)

## Support Vector Machine (SVM)

Algoritmo que busca encontrar um hiperplano ótimo que maximize a margem entre as classes, transformando dados não linearmente separáveis.

- **Funcionamento:** Encontra o hiperplano com a maior margem possível.
- **LinearSVC:** Implementação eficiente para grandes conjuntos de dados.




### Vantagens

- Eficaz em espaços de alta dimensão
- Bom para classificação binária
- Boa generalização

### Desvantagens

- Escala mal com grandes datasets
- Requer escalonamento de features

# 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**

Concatenação dos 8 arquivos CSV do dataset CIC-IDS2017

**2 Análise Exploratória**

Verificação da distribuição de classes e identificação de valores ausentes/infinitos

**3 Pré-processamento**

Limpeza, padronização e tratamento de dados inconsistentes

**4 Agrupamento**

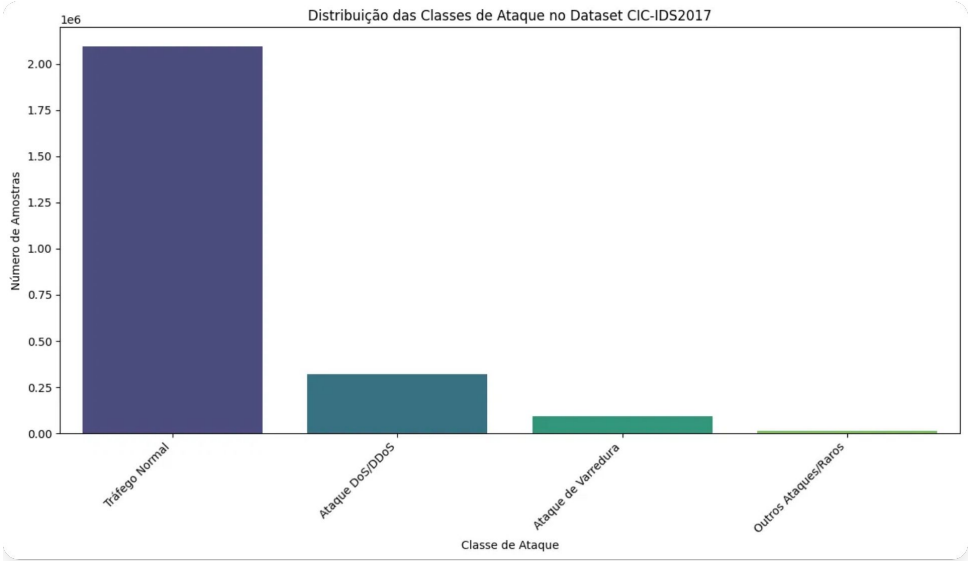
Agrupamento de rótulos em categorias mais amplas para balanceamento

### Distribuição Original

O dataset original apresenta grande desbalanceamento, com predominância de tráfego normal (BENIGN) e diversos tipos de ataques com frequências variadas.

Principais desafios:

- 15 classes diferentes de tráfego
- Desbalanceamento extremo (alguns ataques raros)
- Grande volume de dados (2.8 milhões de registros)




### Agrupamento de Classes

Estratégia de agrupamento em 4 categorias principais:

**Tráfego Normal:** BENIGN

**Ataque DoS/DDoS:**

DoS Hulk, DDoS, DoS GoldenEye, DoS slowloris, DoS Slowhttptest

**Ataque de Varredura:** PortScan

**Outros Ataques/Raros:**

FTP-Patator, SSH-Patator, Bot, Web Attack, Infiltration, Heartbleed

# Metodologia

## Preparação Final dos Dados

**1 Divisão**

Separação em conjuntos de treino (70%) e teste (30%) com estratificação

**2 Escalonamento**

Padronização das features com StandardScaler (média 0, desvio padrão 1)

**3 Seleção**

Seleção das características mais relevantes com SelectKBest

**4 Balanceamento**

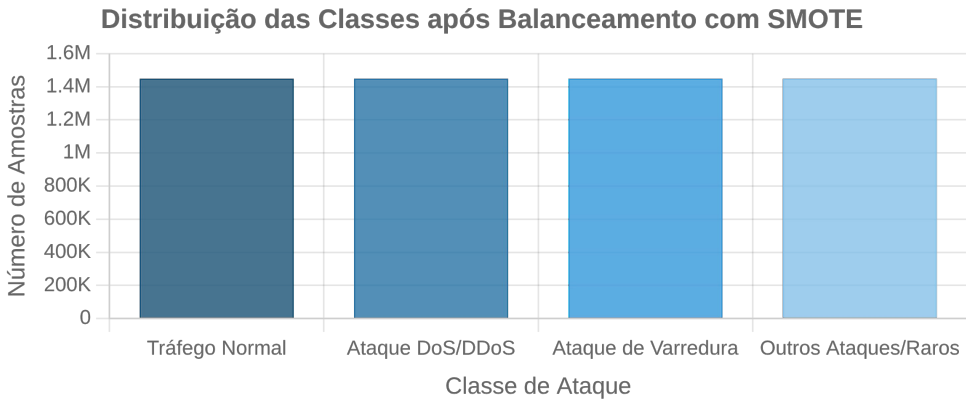
Aplicação de SMOTE para equalizar as classes

### Balanceamento com SMOTE

O SMOTE (Synthetic Minority Over-sampling Technique) cria amostras sintéticas para as classes minoritárias, gerando novos exemplos entre pontos existentes.

Benefícios:

- Reduz o viés do modelo para a classe majoritária
- Melhora a capacidade de generalização
- Aumenta a precisão na detecção de classes raras



# Balanceamento com SMOTE

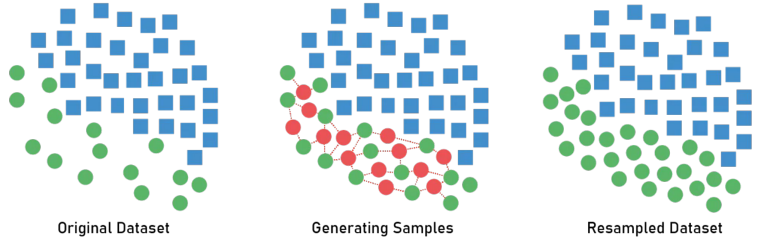
## O que é SMOTE?

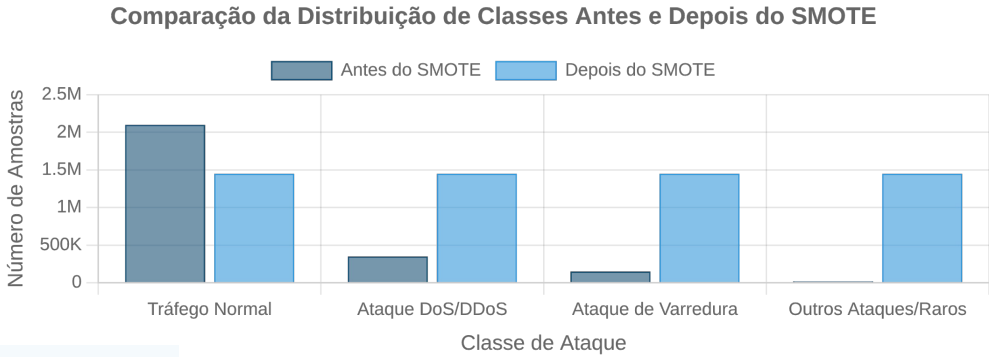
Éuma técnica de balanceamento de dados que cria amostras sintéticas para classes minoritárias, em vez de simplesmente duplicar instâncias existentes.

O algoritmo funciona selecionando exemplos próximos no espaço de características e interpolando novos exemplos sintéticos entre eles.

## Como Funciona

1. Seleciona um exemplo da classe minoritária
2. Encontra seus k vizinhos mais próximos (geralmente k=5)
3. Seleciona aleatoriamente um desses vizinhos
4. Cria um novo exemplo sintético ao longo da linha entre os dois pontos
5. Repete até atingir o balanceamento desejado



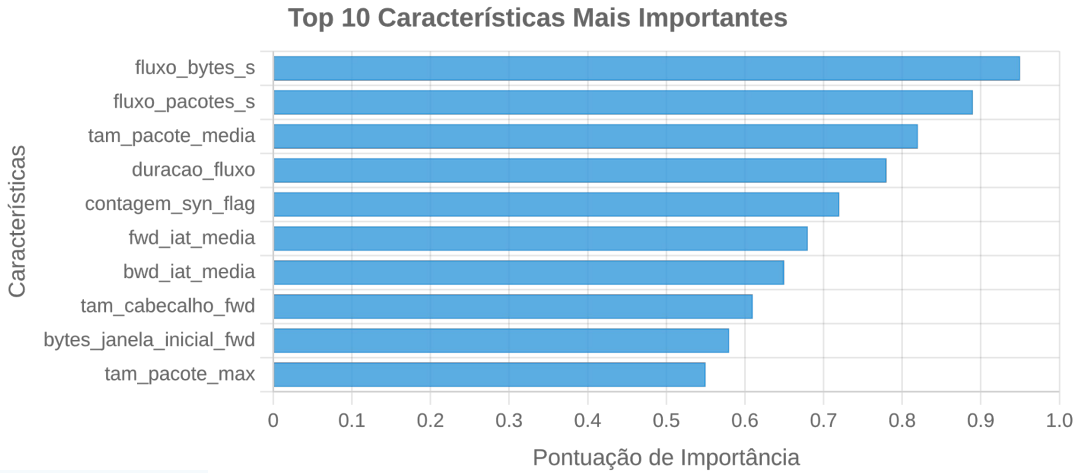

## Benefícios do SMOTE

- ✓ Reduz o viés do modelo
- ✓ Melhora a generalização
- ✓ Aumenta recall em classes raras
- ✓ Evita overfitting

# Seleção de Características

## Por que Selecionar Features?

- **Redução de dimensionalidade:** Diminui a complexidade do modelo
- **Melhoria de desempenho:** Reduz o tempo de treinamento e inferência
- **Prevenção de overfitting:** Elimina características irrelevantes
- **Interpretabilidade:** Facilita a compreensão do modelo




## Características Mais Relevantes

As características mais importantes para a classificação de ataques de rede incluem:

- Duração do fluxo
- Bytes por segundo
- Pacotes por segundo
- Tamanho médio de pacotes
- Contagem de flags
- Tempo entre pacotes
- Tamanho do cabeçalho
- Bytes da janela inicial

# 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

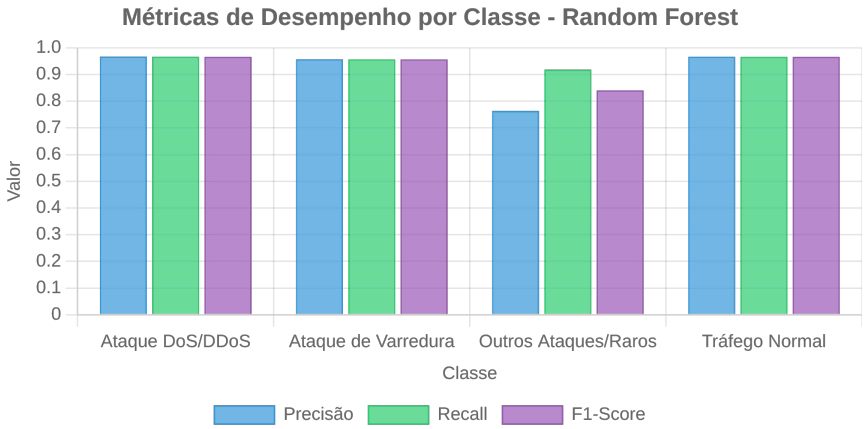
# Resultados: Random Forest

## 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** |

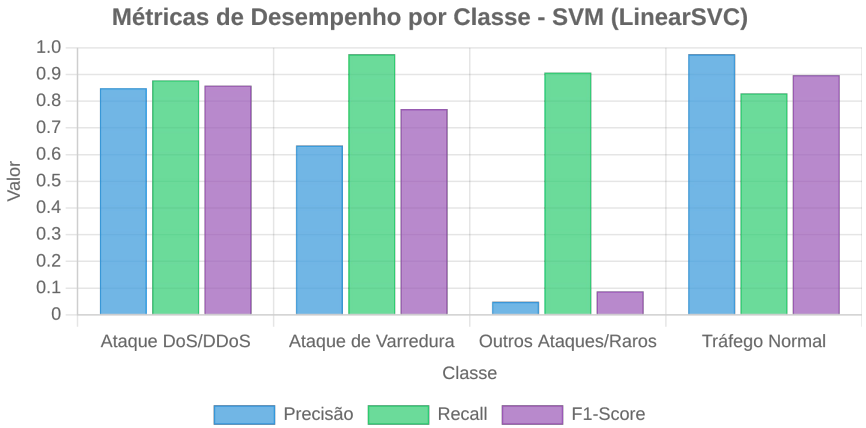
# Resultados: SVM (LinearSVC)

## 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

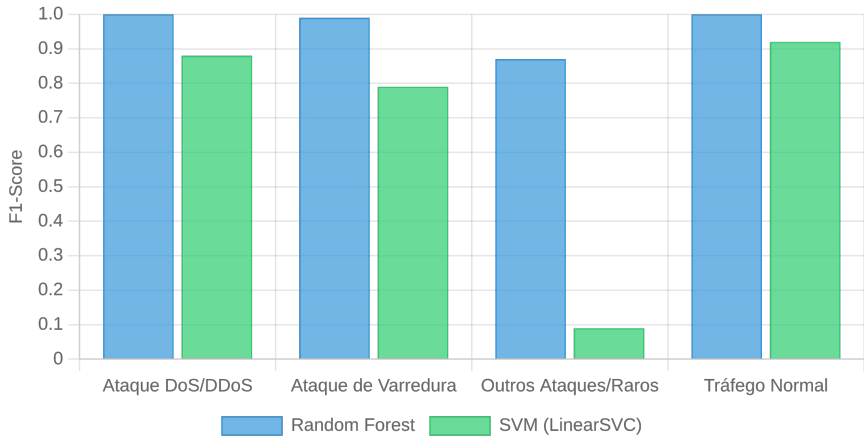
| Tempo de Treinamento | Tempo de Inferência |
|---|---|
| **23.84 minutos** | **0.23 segundos** |

# 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

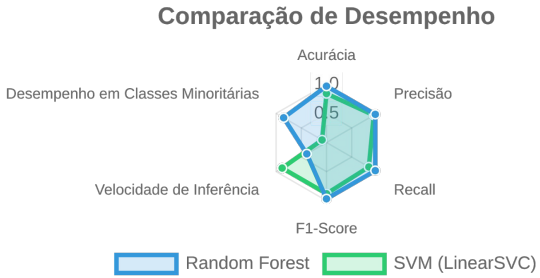
Random Forest foi a melhor escolha para a classificação de ataques no CIC-IDS2017, com acurácia geral de 0.99. Sua capacidade de generalização, impulsionada pelo SMOTE e seleção de características, foi evidente.

## 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.

# Obrigado pela atenção!

Jhonatas Gomes Ribeiro

Instituto Federal de Ciência e Tecnologia do Piauí - IFPI

jhonatasgomes2003@gmail.com