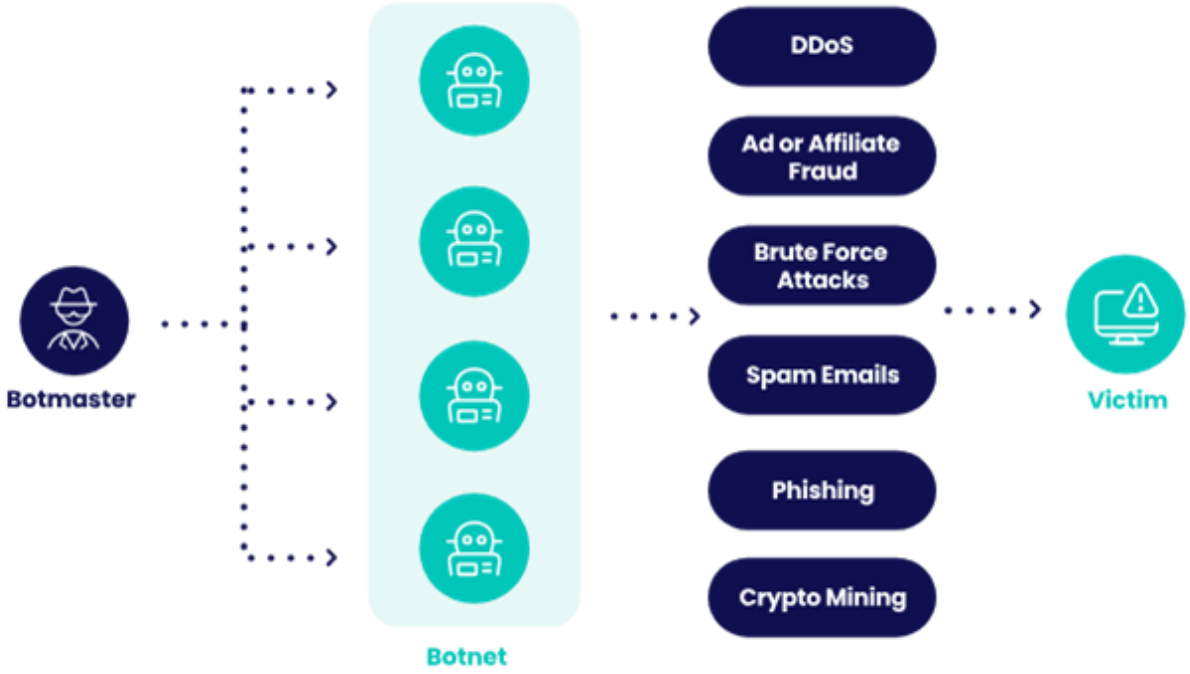
Detecção e Prevenção de Botnets com SVM e Modelagem Preditiva

Isabella de Freitas Nunes

Contexto do problema

- O que é um botnet?



Contexto do problema

- Por que é relevante?
  - Aumento das aplicações de IoT;
  - Importância para a economia global, infraestruturas e serviços de comunicação;
  - ODS 11 da ONU, "Cidades e Comunidades Sustentáveis", estabelece que as cidades devem ser inclusivas, seguras, resilientes e sustentáveis;

Contexto do problema

- MENTORED testbed;

- Objetivo 1: Desenho de uma solução de monitoramento de fluxos de IoT para prevenir ataques e auxiliar na modelagem, detecção e previsão de botnets e ataques DDoS;

- Objetivo 2: Análise de risco pela identificação de botnets formadas por dispositivos IoT e a previsão de ataques DDoS conhecidos e desconhecidos (dia zero);

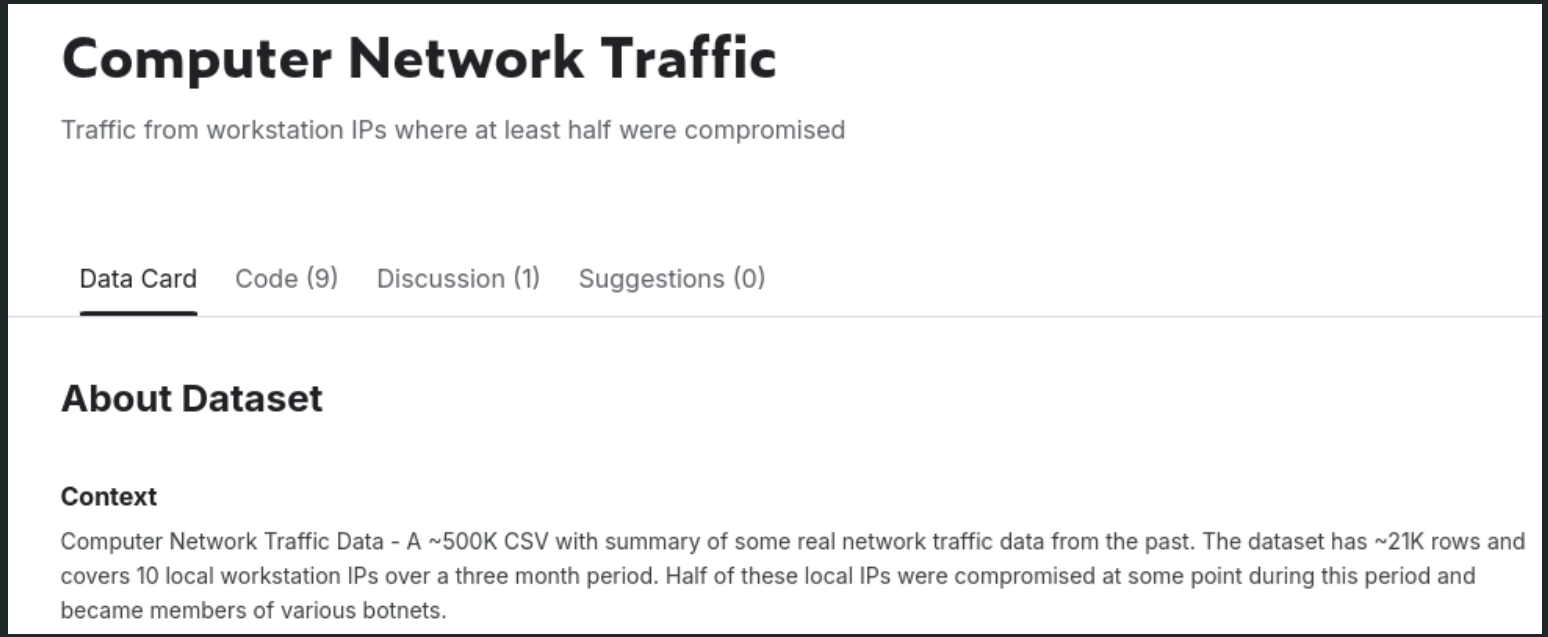
Contexto do problema

- Objetivo 3: Identificação e classificação de comportamentos maliciosos relacionados a ataques DDoS e proposição de solução para detecção e mitigação desses ataques;

- Objetivo 4: Desenho e implementação de um ambiente de experimentação (testbed), com controle de acesso, no qual as soluções propostas possam ser testadas;

Dataset utilizado

- Computer Network Traffic na plataforma kaggle.com



Dataset utilizado

- Dados de tráfego de rede de computadores;
- CSV com resumo de alguns dados reais de tráfego de rede;
- Cerca de 21 mil linhas do tráfego de 10 IPs de dispositivos em um período de três meses. Metade desses IPs locais foram comprometidos em algum momento durante esse período e se tornaram membros de várias botnets;

Dataset utilizado

- Cada linha consiste em quatro colunas:
  - data: aaaa-mm-dd (no formato de ano, mês e dia);
  - l_ipn: IP local (codificado como um inteiro de 0 a 9);
  - r_asn: um inteiro que identifica o ISP remoto;
  - f: contagem de conexões para aquele dia;

Pré-processamento

- Adicionar a coluna **compromised** indicando se um IP estava comprometido em um dado dia;
- Adicionar colunas de datas como: **weekday**, **month**, **is_weekend**;
- Colunas numéricas **r_asn** e **f** foram normalizadas para média 0 e desvio padrão 1;

Algoritmo de classificação

- Algoritmos de aprendizado de máquina supervisionado e de classificação;
- SVM (Support Vector Machine);
- O objetivo do SVM é encontrar um hiperplano que maximize a distância entre as classes, o que é chamado de margem de separação;
- Os pontos que estão na margem de separação são os vetores de suporte;

Algoritmo de classificação

- Kernel RBF (Radial Basis Function): função gaussiana que mapeia dados em um espaço dimensional infinito, sendo um dos kernels mais populares para SVMs;
- Kernel padrão usado no sklearn.svm;

Algoritmo de classificação

- Técnica SMOTE para lidar com balanceamento;
- Essa técnica cria dados sintéticos até que a base de treinamento atinja o equilíbrio de 50% para as classes;

Algoritmo de predição

- Treinamento de um Random Forest Regressor para prever o número de fluxos em um dia;
- O período até 31 de agosto de 2006 é usado para treinar o modelo;
- O período após essa data (setembro de 2006) é usado para testar o modelo;

Resultados

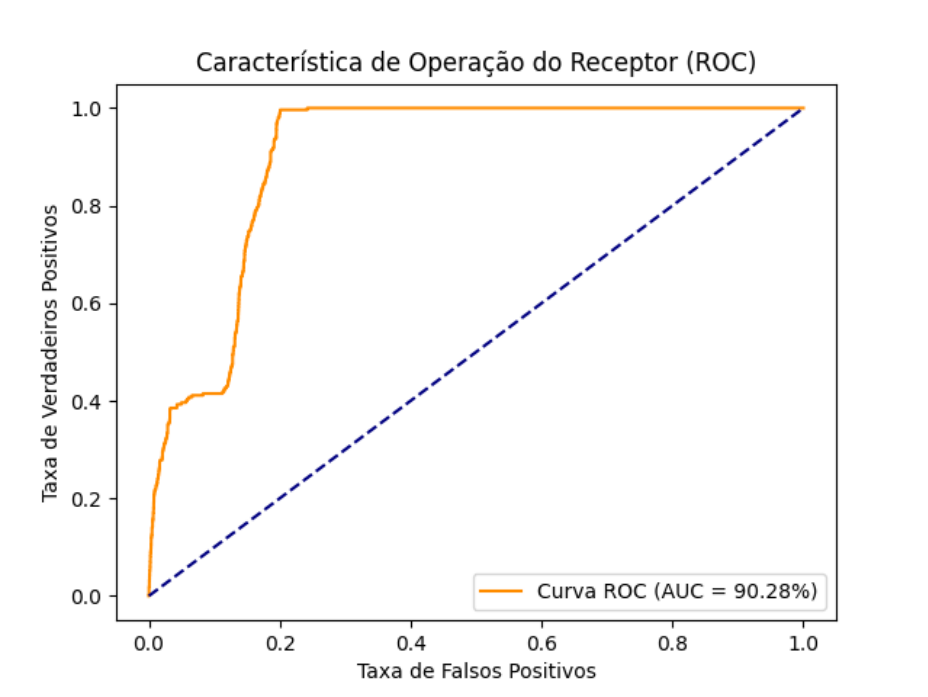
- Matriz de confusão:

Classe Real (Previsão)	**Não Comprometido**	**Comprometido**
Não Comprometido	3103 (verdadeiro negativo)	788 (falso positivo)
Comprometido	1 (falso negativo)	269 (verdadeiro positivo)

Resultados

Métrica	Valor	Interpretação
Acurácia	81.04%	O modelo acertou 81% das classificações
Precisão	25.45%	25% das previsões de IPs comprometidos estavam corretas (alta quantidade de falsos positivos)
Recall	99.63%	Encontrou quase todos os IPs comprometidos
F1 Score	40.54%	Equilíbrio entre precisão e recall é moderado
Fβ Score	29.90%	Baixa precisão

Resultados



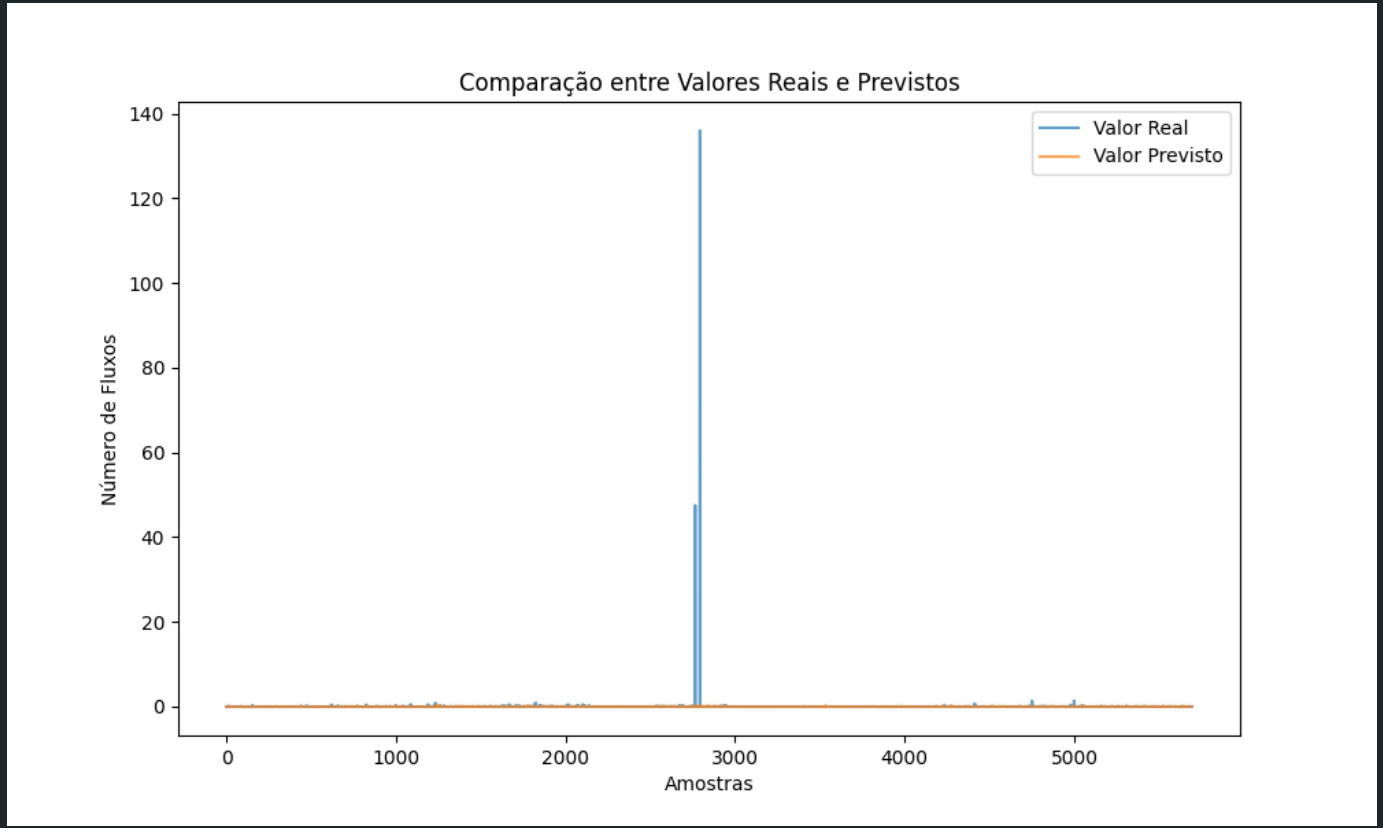
AUC é 90.28%, evidenciando que o modelo é bom em distinguir entre classes (comprometido vs. não comprometido)

Resultados

- Erro Quadrático Médio de 3.65 unidades;
  - Há fortes indícios de outliers;

- Erro Absoluto Médio de 0.04;
  - Modelo teve bom desempenho para a maioria das amostras, mas o Erro Quadrático Médio de 3.65 indica que erros maiores (outliers) aumentaram o erro total;

Resultados


Comparação entre Valores Reais e Previstos
Valor Real
Valor Previsto
Número de Fluxos
Amostras
0
20
40
60
80
100
120
140
0
1000
2000
3000
4000
5000

Resultados

- O modelo é excelente em detectar IPs comprometidos (recall de 99.63%), errando apenas 1 vez ao não identificar um IP comprometido;

- Gera muitos falsos positivos (788), o que reduz a precisão (25.45%);

- O AUC de 90.28% indica que o modelo é bom para distinguir entre as duas classes;

Reflexões

- Baixa precisão do SVM (devido a falsos positivos);
- Dificuldade do modelo de regressão em lidar com outliers;

Código completo

- https://github.com/isadfrn/network-threat-classifier

Obrigada!