Utilização de Técnicas de Machine Learning para Detecção de Botnets

Luis Felipe Bueno da Silva
Orientador: Prof. Dr. Kelton Augusto Pontara da Costa

Tópicos

- Introdução e Fundamentação Teórica
  - Botnets
  - Machine Learning
- Ferramentas utilizadas
- Desenvolvimento
- Resultados
- Conclusão e Trabalhos Futuros

Botnets - Definição

“**Botnets** são redes de computadores comprometidos (**Bots)**, que têm seus recursos utilizados por uma pessoa (**Botmaster)** para cometer vários tipos de crimes virtuais, entre eles ataques de negação de serviço (DDoS), disseminação de vírus, fraudes de cliques para geração de dinheiros em propagandas online, e obtenção de dados pessoais”

Principais arquiteturas e protocolos:

- **Centralizada**
  - IRC
  - HTTP
- **Descentralizada**
  - P2P
- **Híbrida**

Botnets - Centralizadas x Descentralizadas

- Latência
- Facilidade de Uso e Gerenciamento
- Confiabilidade

Botnet - Ciclo de vida

1. Infecção
2. Ambiente de Comando e Controle
3. Ataque
4. Pós-ataque

Botnets - Relevância

- Crescimento no número de computadores pessoais
- Nova tendência surge com a **Mirai,** em 2016
- Explora baixa segurança de dispositivos IoT
- Utilização de senhas fracas

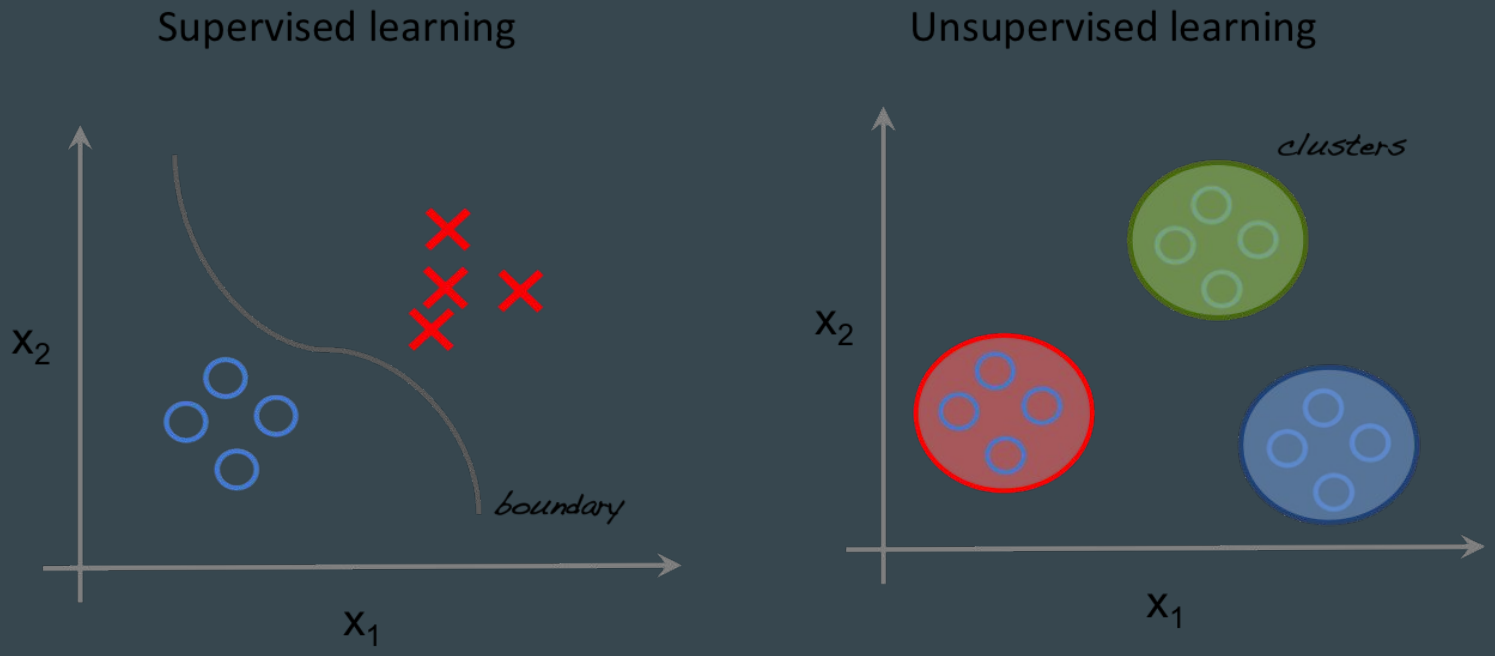
Machine Learning - Definição

Machine Learning é uma subárea da Inteligência Artificial que procura desenvolver algoritmos que sejam capazes de aprender através da observação de padrões nos dados fornecidos, sem que sejam diretamente programados.

Pode ser:

- **Supervisionado**
  - Classificação
  - Regressão
- **Não-supervisionado**
  - Clusterização

Machine Learning - Definição



Objetivo Geral

O objetivo deste trabalho é aplicar algoritmos de Machine Learning capazes de detectar a presença de botnets em uma rede, assim como fazer um estudo de eficiência com base nos resultados obtidos.

Machine Learning - Algoritmos utilizados

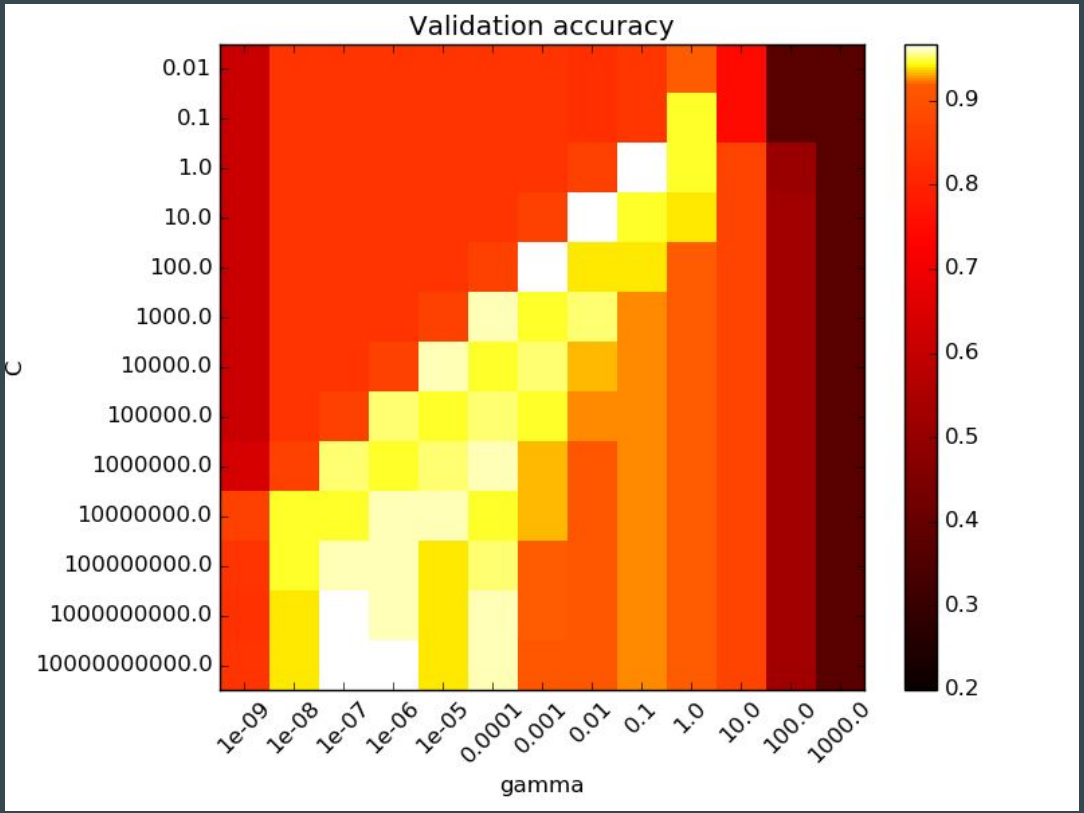
Técnicas mais utilizadas para o estudo supervisionado de Botnets IRC:

- Naive Bayes
- Support Vector Machines
- Árvores de Decisão
- Florestas Aleatórias
- Adaptive Boosting (AdaBoost)

Machine Learning - Recursive Feature Elimination

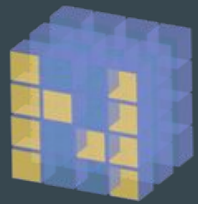
- Busca encontrar um subconjunto ótimo de características
- Atribuição de um ranking de importância
- Elimina característica menos relevante a cada iteração

Machine Learning - Grid Search



Ferramentas

Pandas



NumPy

Desenvolvimento - Base de dados

- Fornecida pela **University of New Brunswick**: ISCX Botnet Dataset
- Três arquivos de pacotes de dados: um para treino e dois para teste
- Lista de IP's maliciosos

Desenvolvimento - Geração de Fluxos

- Um fluxo é definido por IP de origem/destino, Port de origem/destino, e Protocolo
- Programa de código aberto *flowtbag*
- Utilização pela linha de comando
- Processa os arquivos de pacote, separando-os por fluxos e gerando um arquivo no formato CSV
- Além disso, traz mais características como duração de conexão, total de pacotes trocados, etc, totalizando 44 características

Desenvolvimento - Pré-processamento de dados

Inclui tarefas como:

- Filtragem de fluxos pelo protocolo TCP
- Remoção de colunas com dados nulos
- Rotulação
- Geração de características

Desenvolvimento - Pré-processamento de dados

Característica	Fórmula geradora
Total de Bytes	Soma dos bytes enviados em ambas as direções
Total de pacotes	Soma dos pacotes enviados em ambas as direções
Total de Bits	Número total de bytes multiplicado por 8 (1 byte = 8 bits)
Bytes por pacote	Razão entre o Total de Bytes e Total de pacotes
Bytes por segundo	Total de bits dividido pela duração do fluxo
Pacotes por segundo	Total de pacotes dividido pela duração do fluxo
Média de IAT	Média da soma dos valores IAT médios
Bytes por segundo	Total de bits dividido pela duração do fluxo
Média de variância de IAT	Média dos desvio padrão de IAT ao quadrado
Porcentagem de pacotes enviados	Razão entre o número de pacotes enviado na direção *forward* e o número total de pacotes do fluxo
IOPR	Razão entre a quantidade de pacotes na direção *backward* sobre a quantidade da direção *forward*
Média de tamanho de payload	Número de bytes total do fluxo menos a soma de bytes dos headers em ambas as direções, depois dividido pelo número de pacotes

Desenvolvimento - Pré-processamento de dados

```
def flow_total_bytes():
    total_bytes = []
    for index,data in df.iterrows():

        bytes_forward = data['total_fvolume']
        bytes_backward = data['total_bvolume']

        bytes_sum = bytes_forward + bytes_backward

        total_bytes.append(bytes_sum)

    df['total_bytes'] = total_bytes
```

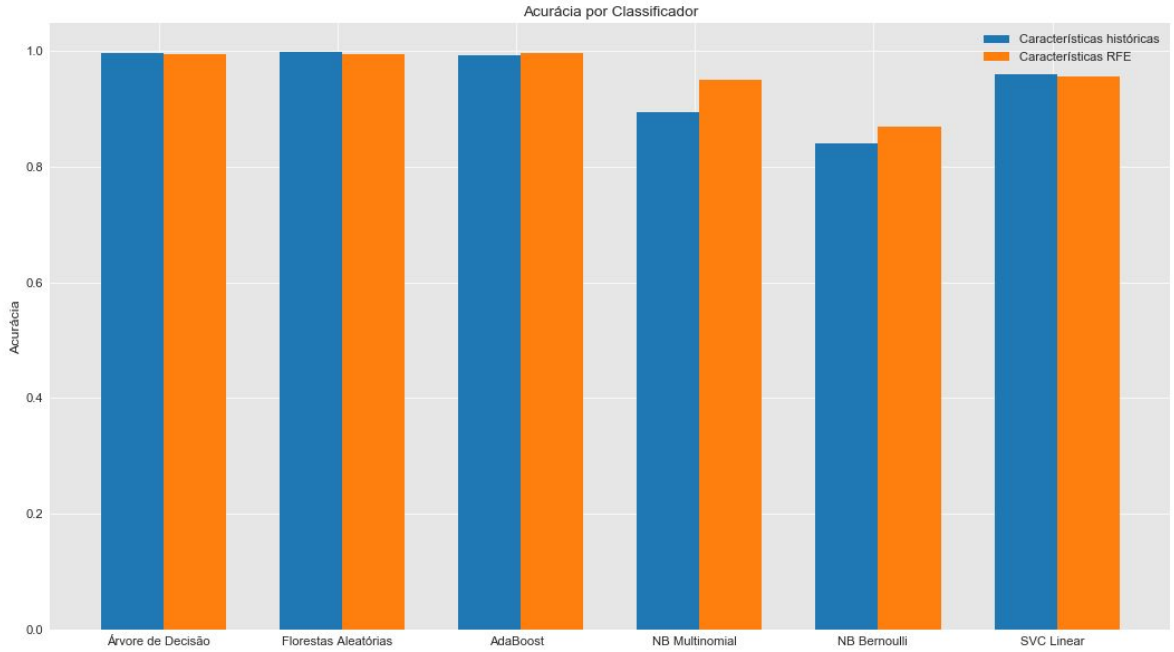
Desenvolvimento - Execução dos algoritmos

Foram utilizadas duas abordagens para encontrar o melhor conjunto de características:

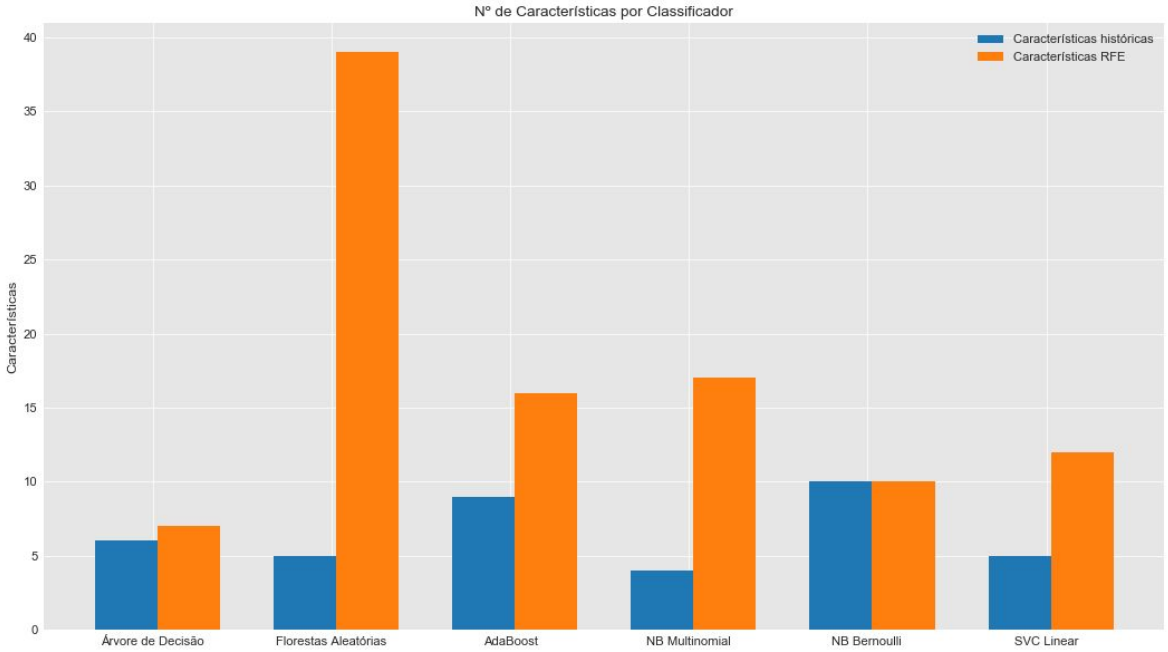
1. Busca de força bruta com base em **dados de pesquisas anteriores**
2. Busca utilizando o algoritmo de **Recursive Feature Elimination** com **todas as características**

Além disso, foi feita a **otimização de hiperparâmetros** do algoritmo de Support Vector Machine utilizando a técnica de **Grid Search**.

Resultados

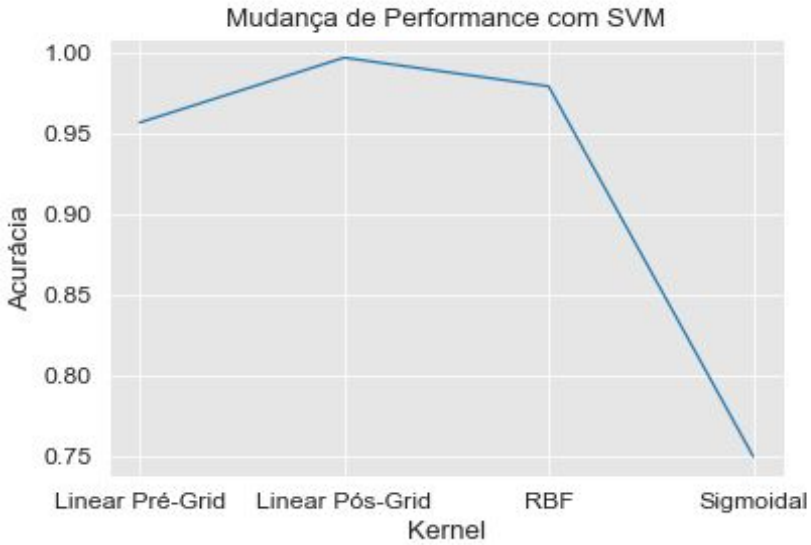

Acurácia por Classificador
Características históricas
Características RFE
Acurácia
1.0
0.8
0.6
0.4
0.2
0.0
Árvore de Decisão
Florestas Aleatórias
AdaBoost
NB Multinomial
NB Bernoulli
SVC Linear

Resultados


Nº de Características por Classificador
Características históricas
Características RFE
Características
0
5
10
15
20
25
30
35
40
Árvore de Decisão
Florestas Aleatórias
AdaBoost
NB Multinomial
NB Bernoulli
SVC Linear

Otimização de Hiperparâmetros - SVM

Kernel	Acurácia	Parâmetros
Linear	0.996604	C:100
RBF	0.978840	C: 100,γ: 0.001
Sigmoidal	0.749477	C: 10,γ: 0.0001, r:0



Conclusão

- Abordagem do RFE representou grande melhora na acurácia nos modelos probabilísticos de Naive Bayes, e resultados similares para os outros
- As características que foram utilizadas nos estudos anteriores, de maneira geral, se fizeram muito presentes nas encontradas pelo RFE
- No entanto, algumas características específicas dos estudos anteriores, como como Duração de Fluxo, tiveram uma representatividade extremamente baixa, não sendo muito efetivas
- Um número considerável de características novas obteve grande representatividade

Trabalhos Futuros

- Utilização de outras características de fluxo de rede;
- Avaliar a eficiência dos métodos e características descritos para outros protocolos de Botnet centralizadas e/ou descentralizadas;
- Aplicação de Redes Neurais Artificiais e algoritmos não-supervisionados;
- Análise de fluxos em tempo real.

Referências

UNIVERSITY OF NEW BRUNSWICK. Botnet dataset. 2018. Disponível em: <http://www-.unb.ca/cic/datasets/botnet.html>. Acesso em: 27 set. 2018.

PEDREGOSA, F. et al. Scikit-learn: Machine learning in Python. Journal of Machine Learning Research, v. 12, p. 2825–2830, 2011.

BEIGI, E. B. et al. Towards effective feature selection in machine learning-based botnet detection approaches. In: IEEE. Communications and Network Security (CNS), 2014 IEEE Conference on. [S.l.], 2014. p. 247–255.

ARNDT, D. Flowtbag. [S.l.]: GitHub, 2011. https://github.com/DanielArndt/flowtbag.

BEIGI, E. B. et al. Towards effective feature selection in machine learning-based botnet detection approaches. In: IEEE. Communications and Network Security (CNS), 2014 IEEE Conference on. [S.l.], 2014. p. 247–255.

MEDIUM. Unsupervised Learning with Python. 2017. Disponível em: <https://towardsdatascience.com/unsupervised-learning-with-python-173c51dc7f03>. Acesso em: 10 nov. 2018.

STACK EXCHANGE. Cross Validated. 2016. Disponível em <https://stats.stackexchange.com/questions/208449/hyper-parameter-optimization-grid-search-issues>. Acesso em 10 nov 2018.