XXXIII Congresso de Iniciação Científica da Unicamp

Desenvolvendo um Modelo de IA para Previsão de Falhas em Redes IPv4

Palavras-Chave: Redes de Computadores; Aprendizado de Máquina; Detecção de Anomalias

Autores/as:
Felipe Akira Nozaki, FT, UNICAMP
Prof. Dr. André F. de Angelis, FT, UNICAMP

1. INTRODUÇÃO

Redes de computadores estão sujeitas a falhas e ataques, como negação de serviço e varreduras de porta. Este trabalho investiga o uso de algoritmos de aprendizado de máquina para detectar tráfego malicioso em redes IPv4, estruturando o problema como uma classificação binária entre tráfego legítimo e não legítimo. Além dos modelos de aprendizado supervisionado, também foi implementado um método estatístico tradicional, permitindo uma comparação entre as diferentes abordagens para a detecção de anomalias.

2. METODOLOGIA

O trabalho foi conduzido em etapas sequenciais, desde o preparo dos dados até a avaliação dos modelos.

2.1 Fase exploratória em ambiente virtual

Inicialmente, com o objetivo de gerar dados próprios para treinar os modelos de aprendizado de máquina, utilizou-se um ambiente controlado no *Oracle VirtualBox*, onde ataques foram simulados entre duas máquinas virtuais. No entanto, a baixa diversidade dos registros e a ausência de tráfego legítimo comprometeram a qualidade do conjunto de dados. Diante dessas limitações, optou-se pela utilização de uma base pública consolidada nas etapas seguintes. Na Figura 1, é possível observar a interface da ferramenta utilizada para simular o ambiente controlado.

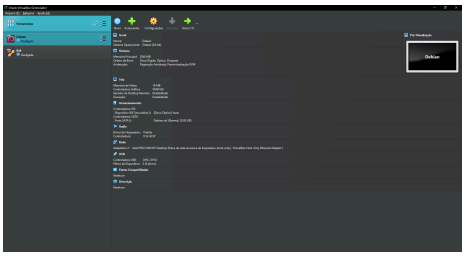

Figura 1 - Interface do Oracle VirtualBox (Fonte: Autor)

2.2 Coleta e preparação dos dados

A base CICIDS2017 foi selecionada por conter registros realistas de tráfego de rede, incluindo diferentes tipos de ataques. Para este trabalho, foram utilizados dois arquivos específicos, relacionados a ataques de negação de serviço distribuído (*DDoS*) e varredura de porta (*Portscan*), somando mais de 500 mil registros. A Figura 2 apresenta a distribuição das classes na variável alvo.

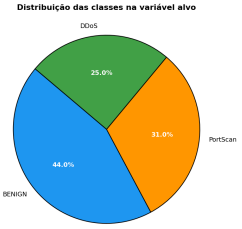

Figura 2 - Distribuição das classes na variável alvo (Fonte: Autor)

Após a seleção dos dados, realizou-se o pré-processamento, que envolveu a remoção de 15% dos registros duplicados e o tratamento de valores infinitos e ausentes, deixando os dados prontos para uso. Por fim, a variável alvo foi convertida para o formato binário, sendo 0 para tráfego benigno e 1 para tráfego malicioso.

2.3 Redução de atributos

Para simplificar o conjunto de dados e facilitar o trabalho dos modelos, foi necessário reduzir a quantidade de atributos utilizados. O conjunto original possuía 79 atributos. Inicialmente, foram removidos 10 atributos constantes, que apresentavam o mesmo valor em todos os registros, e 32 atributos altamente correlacionados, restando 37 atributos no conjunto final.

Além disso, foi realizada uma análise exploratória com PCA (Análise de Componentes Principais) para verificar o quanto seria possível reduzir ainda mais a dimensionalidade sem perder informações importantes. O resultado mostrou que 22 componentes principais resumiam aproximadamente 95% da variação dos dados, servindo de base para escolher os cenários de teste nas etapas seguintes.

2.4 Construção das pipelines

Depois do tratamento e da redução dos atributos, as etapas do experimento foram organizadas em pipelines, ou seja, em sequências automáticas que processam e avaliam os dados do começo ao fim. Seis modelos de classificação foram testados: árvore de decisão, floresta randômica, KNN (vizinhos mais próximos), regressão logística, SVM (máquinas de vetor de suporte) e MLP (rede neural multicamadas).

Cada modelo foi avaliado em cinco cenários distintos, definidos pela combinação entre método de seleção de atributos (SelectKBest, que escolhe as variáveis mais relevantes com base em testes estatísticos, ou RFE, que elimina atributos menos importantes de forma iterativa) e a quantidade de atributos (10 ou 22), além do uso direto dos 37 atributos finais sem seleção.

- Uso dos 37 atributos finais (sem seleção),
- Seleção dos 22 mais relevantes com SelectKBest,
- Seleção dos 10 mais relevantes com SelectKBest,
- Seleção dos 22 mais relevantes com RFE,
- Seleção dos 10 mais relevantes com RFE.

Essas cinco abordagens permitiram comparar o desempenho dos modelos com diferentes algoritmos de seleção de atributos, e também com diferentes quantidades de atributos. Em cada pipeline, os dados eram normalizados, depois passavam pela seleção dos atributos e, por fim, pelo modelo de classificação. Essa organização garantiu condições iguais de teste para todos os modelos, permitindo uma comparação justa dos resultados.

2.5 Treinamento e avaliação inicial

As pipelines foram aplicadas ao conjunto CICIDS2017, pré-processado anteriormente. Utilizou-se validação cruzada estratificada para garantir a mesma proporção de classes em cada divisão de treino e teste. Em cada rodada, foram calculadas as principais métricas de avaliação: acurácia, precisão, revocação, F1-score e tempo de execução.

Essa etapa teve como objetivo obter uma visão inicial do desempenho de cada modelo e método de seleção de atributos, ajudando a escolher as configurações mais promissoras para as próximas fases do experimento.

2.6 Avaliação externa

Para avaliar a capacidade dos modelos de generalizar para novos dados, foi realizada uma avaliação externa. Nessa etapa, as pipelines foram treinadas com todo o conjunto CICIDS2017 e testadas no CICIDS2018, que contém registros semelhantes, como força bruta, ataques de negação

de serviço e ataques de negação de serviço distribuídos.

Esse teste em uma base diferente simulou uma situação mais próxima do mundo real, permitindo observar como os modelos lidam com padrões que não estavam presentes no treinamento.

As mesmas métricas anteriores foram utilizadas para avaliar o desempenho, e os resultados dessa etapa serviram como base para selecionar as configurações mais adequadas para a fase final do experimento.

2.7 Ajuste de hiperparâmetros

Após identificar as melhores combinações de modelo e seleção de atributos, foi realizado o ajuste de hiperparâmetros apenas nas pipelines mais promissoras, por exemplo, apenas nas de 22 atributos, já que as versões com 37 não apresentaram ganhos relevantes e as com 10 reduziram o desempenho.

O ajuste foi feito por meio da ferramenta *GridSearchCV*, que testa diversas combinações de parâmetros automaticamente. Foram ajustados, por exemplo, a profundidade máxima e o número de árvores na Floresta Randômica, os níveis de regularização na Regressão Logística e SVM, e as configurações de camadas no modelo MLP.

2.8 Implementação do método tradicional

Para fins comparativos, também foi implementado um método estatístico simples, que busca detectar anomalias analisando se certas variáveis se afastam muito da média esperada, com base em desvios-padrão. A ideia é classificar como malicioso quando o tráfego apresentar atributos que ultrapassem os valores esperados da média. Essa abordagem foi aplicada aos mesmos dados utilizados pelos modelos de IA, possibilitando uma comparação direta entre as duas estratégias.

3. RESULTADOS E DISCUSSÕES

3.1 Avaliação inicial

Floresta Randômica e Árvore de Decisão apresentaram os melhores resultados, com F1-score acima de 99,98% em quase todos os cenários, como pode ser visualizado na Tabela 1. MLP e KNN também se saíram bem, mas o KNN teve queda de desempenho com poucos atributos.

Tanto o SelectKBest quanto o RFE funcionaram bem ao selecionar 22 atributos. No entanto, a redução para 10 comprometeu o desempenho de alguns modelos, especialmente KNN e Regressão Logística.

Esses altos índices de F1-score podem ser explicados, em parte, pela relativa facilidade de padronização dos dados presentes no CICIDS2017, o que favorece o desempenho dos modelos durante a avaliação inicial.

Tabela 1 - F1-score (%) dos modelos testados

Modelo	Atributos	SelectK Best (22)	SelectK Best (10)	RFE (22)	RFE (10)
Árvore de decisão	99,98%	99,98%	99,79%	99,98%	99,97%
Regressão Logística	99,22%	97,19%	94,19%	98,42%	95,75%
Floresta Randômica	99,98%	99,98%	99,79%	99,99%	99,98%
KNN	99,95%	99,96%	89,43%	99,95%	99,91%
SVM	99,44%	97,28%	94,42%	98,23%	95,96%
MLP	99,92%	99,73%	99,64%	99,90%	99,60%

3.2 Avaliação externa

Na avaliação com os dados do CICIDS2018, os modelos tiveram mais dificuldade nos ataques de força bruta. A Floresta Randômica teve o pior resultado, com F1-score em torno de 49%, enquanto KNN, MLP e Árvore de Decisão se saíram melhor, com resultados acima de 74% (Figura 4).

Nos ataques DoS, o desempenho foi bem mais alto. A maioria dos modelos superou 90% de F1-score, com destaque para Árvore de Decisão, Floresta Randômica e KNN (Figura 5).

Nos ataques DDoS, o desempenho continuou alto, repetindo o padrão observado nos ataques DoS (Figura 6). Esse bom resultado era

esperado, já que os modelos foram treinados com dados semelhantes, especialmente com ataques DDoS.

Em relação ao tempo de execução, a Árvore de Decisão foi a mais rápida, seguida da Floresta Randômica. O KNN foi o mais lento, passando de 240 segundos, enquanto SVM e Regressão Logística foram rápidos, mas com desempenho inferior (Figura 7).

De forma geral, os modelos baseados em árvore mostraram o melhor equilíbrio entre desempenho e tempo de resposta.

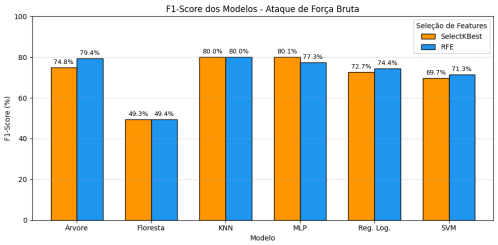

Figura 4 - F1-score por modelo com dados de força bruta (Fonte: Autor)

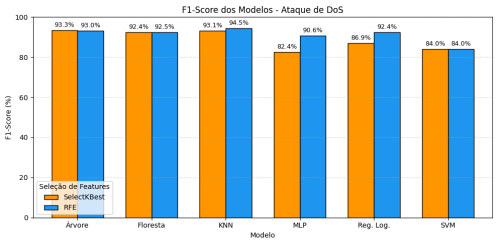

Figura 5 - F1-score por modelo com dados de ataques DoS (Fonte: Autor)

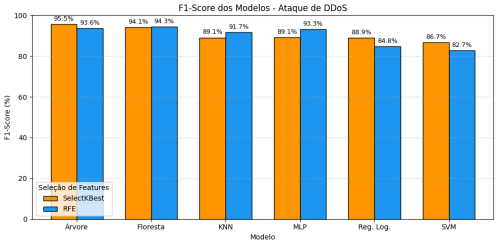

Figura 6 - F1-score por modelo com dados de ataques DDoS (Fonte: Autor)

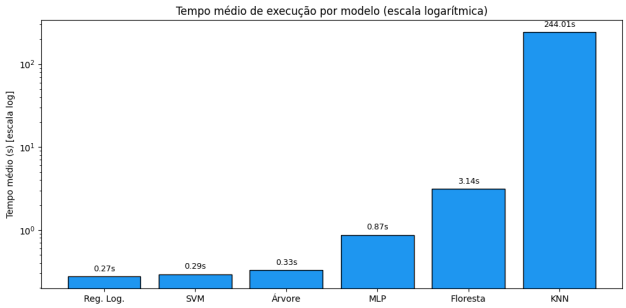

Figura 7 - Tempo de execução médio por modelo (Fonte: Autor)

3.3 Ajuste de hiperparâmetros

O ajuste de hiperparâmetros foi feito para melhorar o desempenho dos modelos, com exceção do KNN, que foi excluído por ser muito lento. O processo foi demorado, especialmente para a Floresta Randômica com RFE (mais de 2400s) e o MLP (acima de 1000s), devido à validação cruzada com até 72 combinações por modelo.

Apesar do esforço, a maioria dos modelos manteve configurações próximas das iniciais, indicando que os valores padrão já estavam bem ajustados ao problema.

Os tempos médios de execução de cada modelo podem ser vistos na Figura 8.

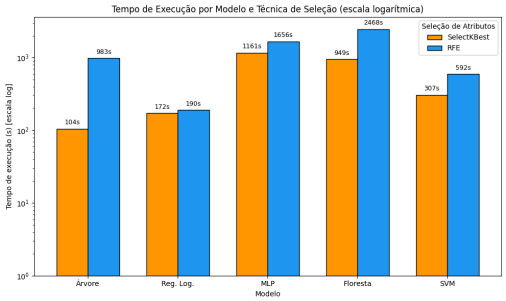

Figura 8 - Tempo de execução do ajuste de hiperparâmetros por modelo (Fonte: Autor)

3.4 Avaliação do método tradicional

O método tradicional, baseado em desvios-padrão, teve resultados bastante inferiores aos modelos de IA. No ataque de força bruta, onde as variáveis apresentavam maior correlação linear com a classe, o desempenho foi o “menos pior”, alcançando um F1-score de 49,16%. Em seguida, no ataque DoS, o F1-score caiu para 18,31%, e no ataque DDoS, que tinha as variáveis menos correlacionadas com a classe, o desempenho foi praticamente nulo, com F1-score de apenas 4,66%.

Essa queda progressiva mostra que o método funcionou um pouco melhor quando havia uma relação mais direta entre os dados e o tipo de ataque, mas teve grande dificuldade em lidar com padrões mais complexos ou menos lineares.

CONCLUSÕES

Este trabalho avaliou diferentes modelos de aprendizado de máquina para detecção de tráfego malicioso em redes IPv4, utilizando um conjunto público com ataques variados. Os modelos baseados em árvores, especialmente Floresta Randômica e Árvore de Decisão, se destacaram por combinar alto desempenho com baixo tempo de execução. Já o KNN e os modelos lineares apresentaram limitações, seja no tempo ou na precisão.

A seleção de atributos foi fundamental para reduzir a complexidade dos dados. O SelectKBest e o RFE permitiram bons resultados com menos variáveis, sendo que o SelectKBest se mostrou mais simples e eficiente, especialmente ao utilizar 22 atributos.

O ajuste de hiperparâmetros exigiu bastante tempo de execução e resultou em poucas melhorias, reforçando que as configurações padrão já estavam bem adequadas ao problema.

Além disso, foi avaliado um método tradicional baseado em desvios-padrão. Seu desempenho foi bem inferior aos modelos de IA, com F1-scores decrescentes em ataques com menor correlação linear: 49,16% (força bruta), 18,31% (DoS) e 4,66% (DDoS). Isso mostra a dificuldade do método em lidar com padrões mais complexos.

A decisão de treinar os modelos apenas com ataques DDoS e PortScan teve impacto na generalização: os modelos tiveram dificuldade em identificar ataques diferentes dos vistos no treinamento. Isso reforça a importância de treinar com uma variedade de ameaças.

No geral, a combinação do modelo Floresta Randômica com SelectKBest utilizando 22 atributos foi a mais eficiente, servindo como referência para o desenvolvimento de sistemas de detecção de anomalias em redes de computadores.

Como continuidade, recomenda-se ampliar o conjunto de dados com outros tipos de ataques e testar abordagens mais avançadas, como redes neurais profundas e técnicas de detecção em tempo real.

BIBLIOGRAFIA

KUROSE, James F.; ROSS, Keith W. Redes de computadores e a internet: uma abordagem top-down. 6. ed. 2013.

CANADIAN INSTITUTE FOR CYBERSECURITY. CICIDS2017 Dataset. University of New Brunswick, 2017. Disponível em: https://www.unb.ca/cic/datasets/ids-2017.html. Acesso em: 11 dez. 2024.

CANADIAN INSTITUTE FOR CYBERSECURITY. CICIDS2018 Dataset. University of New Brunswick, 2018. Disponível em: https://www.unb.ca/cic/datasets/ids-2018.html. Acesso em: 02 fev. 2025.

MULLER, Andreas C.; GUIDO, Sarah. Introduction to machine learning with Python: a guide for data scientists. Sebastopol: O'Reilly Media, 2016.

APOIO

Este trabalho foi desenvolvido com o apoio do Programa Institucional de Bolsas de Iniciação Científica (PIBIC).

AGRADECIMENTOS

Agradeço ao professor orientador, Prof. Dr. André F. de Angelis, pelo acompanhamento e orientação durante o projeto, e à minha família pelo incentivo e apoio ao longo desta jornada.