Detecção de ataques em fluxos de rede utilizando Naive Bayes

Carolina G. de A. B. dos Santos, Eduardo A. P. V. de Oliveira, Isabela M. de M. Nascimento,
Karina L. de Oliveira, Kauanny K. D'A. A. e Barros
Centro de Informática (CIn), Universidade Federal de Pernambuco (UFPE)
Recife, Brasil
{cgabs, eapvo, immn, klo, kkdab}@cin.ufpe.br

***Abstract*—Essa pesquisa utiliza os modelos de aprendizado de máquina para identificar padrões em fluxos de rede, classificando atividades como benignas ou como ataques cibernéticos populares por meio do modelo Naive Bayes. A análise exploratória avalia inconsistências e correlações entre variáveis, enquanto métricas como precisão, recall e acurácia medem o desempenho do modelo. Implementado em Python no Google Colaboratory, o projeto explora o potencial da Inteligência Artificial na cibersegurança.**

***Index Terms*—Cibersegurança, Naive Bayes, classificação, detecção de intrusões.**

I. Introdução

A crescente digitalização dos processos e a interconexão global através da internet têm trazido inúmeros benefícios, mas também expõem organizações e indivíduos a crescentes ameaças cibernéticas, podendo causar prejuízos financeiros e operacionais significativos. Nesse contexto, a detecção de ataques cibernéticos é essencial para proteger redes e sistemas. Esste projeto busca desenvolver um modelo baseado no algoritmo Naive Bayes para detecção de ataques em fluxos de rede, utilizando aprendizado de máquina para identificar padrões anômalos e diferenciar comportamentos legítimos de maliciosos, contribuindo para o fortalecimento da cibersegurança.

II. Objetivo(s)

O objetivo deste projeto é realizar a análise do dataset *CIC-IDS2017* [1], que contém informações sobre fluxos de rede, representando tanto dados benignos como ataques populares. Essa análise busca compreender as características dos dados, identificando padrões relevantes que auxiliem na detecção de intrusões. Além disso, será conduzida uma análise exploratória dos dados para:

- Identificar e tratar possíveis inconsistências ou valores ausentes no dataset;
- Analisar a distribuição das classes e o equilíbrio entre elas;
- Investigar a correlação entre as variáveis e sua relevância para tarefa de classificação.

Posteriormente, um modelo de aprendizado de máquina será treinado com o objetivo de realizar a classificação dos dados de teste. A análise de desempenho será realizada utilizando métricas como precisão, recall, F1-score e acurácia, permitindo uma avaliação detalhada da eficácia do modelo no contexto do problema.

III. Justificativa

A área de cibersegurança está ganhando cada vez mais espaço no mundo atual, impulsionada pelo aumento exponencial de dispositivos conectados e pela crescente sotisficação de ataques cibernéticos. Com organizações enfrentando desafios significativos para proteger suas infraestruturas, dados e operações, torna-se essencial explorar abordagens inovadoras para detecção e mitigação de ameaçsas.

Nesse contexto, a escolha da temática deste projeto justifica-se pela relevância de plicar técnicas de *Inteligência Artificial* (IA) à cibersegurança, buscando soluções eficientes, rápidas e escaláveis. Modelos de aprendizado de máquina, como o *Naive Bayes* [2], tem se mostrado particularmente úteis devido à sua simplicidade, rapidez de treinamento e eficiência em tarefas de classificação, mesmo em cenários com grandes volumes de dados.

Ao analisar o comportamento do modelo *Naive Bayes* por meio de métricas como precisão, recall, F1-score e acurácia, este projeto visa demonstrar como a IA pode ser aplicada para identificar padrões em fluxos de rede e classificar atividades como benignas ou maliciosas. Essa abordagem possibilita não apenas um entendimento mais profundo do potencial desse modelo específico, mas também oferece uma avaliação prática de como soluções baseadas em IA podem ser aplicadas na cibersegurança.

IV. Metodologia

Inicialmente, será realizado a análise exploratória dos dados do dataset *CIC-IDS2017* para compreensão de como as variáveis se comportam e vizualisar suas distribuições. Seguido dessa etapa, será utilizado o modelo classificador *Naive Bayes* para classificar os dados como benignos ou como algum ataque de rede popular. Posteriormente, será realizado a análise das métricas do modelo para visualizar como ele se performa neste cenário.

Para isso, todo projeto será desenvolvido na linguagem *Python*, em um ambiente de desenvolvimento do *Google Colaboratory*. As principais bibliotecas utilizadas serão: *Numpy, Pandas, Os, Re, Google, Matplotlib, Seaborn, Scikit learn* e *Tqdm*.

A. Dataset

O *CIC-IDS2017* é um conjunto de dados amplamente utilizado em sistemas de detecção de intrusão (IDS) e sistemas de prevenção de intrusão (IPS), ele foi desenvolvido com o objetivo de fornecer dados realistas para a avaliação de abordagens de detecção de intrusão baseadas em anomalias. Este dataset possui o tráfego de rede beningno e ataques comuns, representando dados do mundo real, capturados por meio de pacotes de captura de rede (PCAPs), dentre as informações detalhadas de tráfegos encontram-se IPs de origem e destino, portas de origem e destino, protocolos e tipos de ataque. Além disso, os fluxos de rede são etiquetados com base em métricas como timestamps, o que facilita a análise e a classificação de eventos.

O dataset foi capturado entre os dias 3 e 7 de julho de 2017, com um foco particular em um dia típico de tráfego (segunda-feira) e a execução de diversos ataques nos dias subsequentes. Durante esse período, ataques como Brute Force FTP, Brute Force SSH, DoS, Heartbleed, Web Attack, Infiltração, Botnet e DDoS foram realizados, permitindo a análise de uma variedade de cenários de segurança. Esses ataques foram realizados em diferentes horários, tanto pela manhã quanto à tarde, durante dias específicos, garantindo uma cobertura abrangente e representativa dos diversos tipos de ameaça cibernética.

Além disso, o *CIC-IDS2017* inclui uma topologia de rede completa, com diversos dispositivos e sistemas operacionais, abrangendo tanto as redes dos atacantes quanto das vítimas. A comunicação entre diferentes máquinas, incluindo servidores e clientes, foi capturada, representando uma rede variada com sistemas operacionais como Windows, Ubuntu e Mac OS. Ele possui dados de tráfego de protocolos comuns em redes modernas, como HTTP, HTTPS, FTP, SSH e e-mail. Isso oferece uma visão detalhada dos principais tipos de comunicação que ocorrem em ambientes de rede.

Completamente etiquetado, o dataset oferece metadados detalhados, como hora de ocorrência, tipo de ataque, fluxos e rótulos, permitindo que pesquisadores e engenheiros de segurança realizem análises aprofundadas. A estrutura do dataset é composta por arquivos no formato CSV, contendo mais de 80 features extraídas dos fluxos de rede, o que torna o conjunto de dados ideal para análise utilizando técnicas de aprendizado de máquina e aprendizado profundo.

B. Processamento do Dataset

O processamento do dataset é uma etapa essencial para garantir a qualidade dos dados utilizados no treinamento do modelo. Inicialmente, o dataset, que está no formato CSV, será carregado para o ambiente de trabalho. Após o carregamento, será realizado um ajuste nos nomes das colunas, removendo espaços em branco. Esse procedimento padroniza a nomenclatura, facilitando as operações subsequentes e reduzindo possíveis erros durante o processamento.

Em seguida, será realizada a limpeza dos dados para garantir que apenas informações relevantes e consistentes sejam utilizadas. Essa etapa inclui a remoção de registros duplicados, que poderiam introduzir redundância e vieses no modelo. Também serão tratados os valores ausentes (NaN, Null ou NA) e não finitos (como infinito ou -infinito), que podem comprometer a integridade das análises. Esses registros problemáticos serão excluídos para evitar impactos negativos no desempenho do modelo.

Após a limpeza, será conduzida uma análise exploratória dos dados, com o objetivo de compreender a distribuição das variáveis, identificar padrões importantes e detectar possíveis outliers. Essa análise incluirá a criação de visualizações gráficas que facilitem a interpretação dos dados. Posteriormente, o dataset será dividido em dois subconjuntos distintos: dados de treino e de teste. Além disso, será realizada uma análise de correlação entre as features para identificar variáveis que apresentam alta correlação entre si. Tais variáveis redundantes serão descartadas, uma vez que podem dificultar o aprendizado do modelo ao introduzir informações repetitivas.

Outra etapa crucial será a normalização dos dados, ajustando todas as variáveis para uma escala comparável. Esse procedimento é especialmente importante para algoritmos como o Naive Bayes, que podem ser sensíveis à escala das features. A normalização garantirá que nenhuma variável tenha peso excessivo no treinamento do modelo. Por fim, o modelo de Naive Bayes será treinado utilizando o conjunto de dados pré-processado. Essa etapa final integrará todo o trabalho realizado previamente, permitindo a construção de um modelo eficiente e preciso. O objetivo principal é obter o melhor desempenho possível, garantindo a confiabilidade das previsões realizadas pelo modelo.

C. Teorema de Bayes

O Teorema de Bayes, um princípio fundamental na teoria das probabilidades e na estatística, é utilizado para calcular a probabilidade de um evento com base em informações prévias relacionadas a ele. Ele descreve como ajustar uma estimativa inicial ao considerar novas evidências ou dados observados, produzindo uma probabilidade revisada mais precisa.

Esse teorema leva o nome do matemático e pastor inglês Thomas Bayes (1701-1761), que desenvolveu ideias relacionadas à probabilidade condicional e sua aplicação em distribuições binomiais. Embora seus trabalhos só tenham ganhado destaque após sua morte, a contribuição de Bayes fundamenta muitos dos avanços na estatística moderna, especialmente no campo da inferência bayesiana. Uma das fórmulas mais conhecidas do Teorema de Bayes é:

$$P(A|B) = \frac{P(B|A) \times P(A)}{P(B)}$$

Onde $P(A \mid B)$ é a probabilidade de A ocorrer dado que B ocorreu, $P(B \mid A)$ é a probabilidade de B dado A, $P(A)$ é a probabilidade inicial de A, e $P(B)$ é a probabilidade de B.

O Teorema de Bayes tem aplicações amplas e práticas. Ele é essencial em áreas como aprendizado de máquina, diagnóstico médico, análise de risco e até mesmo em sistemas de recomendação. Um exemplo clássico é o uso em classificadores probabilísticos, como o Classificador Naive Bayes,

que calcula a probabilidade de uma instância pertencer a uma categoria específica com base em seus atributos.

Além disso, o Teorema de Bayes é a base da abordagem bayesiana na estatística, que busca incorporar conhecimento prévio em análises, tornando-a uma ferramenta poderosa para problemas em que há incerteza ou dados limitados. Essa metodologia permite atualizações iterativas e dinâmicas das probabilidades à medida que novas informações surgem, ampliando sua utilidade em diversas disciplinas.

D. Classificador do Naive Bayes

O Classificador Naive Bayes é um dos algoritmos mais conhecidos e amplamente utilizados no aprendizado de máquina, tanto no meio acadêmico quanto no mercado. Ele apresenta uma abordagem direta e eficiente para resolver problemas de classificação, com base em princípios estatísticos sólidos.

O algoritmo é inspirado no trabalho do matemático Thomas Bayes e aplica o Teorema de Bayes para realizar previsões. A palavra "naive" (ingênuo) refere-se à suposição simplificadora de que as características (ou atributos) dos dados são completamente independentes entre si. Embora essa premissa possa não ser verdadeira em muitos casos do mundo real, ela permite que o algoritmo seja computacionalmente eficiente e fácil de implementar.

Outro pressuposto importante do Naive Bayes é que todas as variáveis (features) têm igual relevância no resultado final. Quando essa condição não é atendida, o desempenho do algoritmo pode ser prejudicado, tornando-o menos adequado para situações onde as dependências entre atributos são significativas.

Apesar de suas limitações, o Naive Bayes é amplamente reconhecido por sua simplicidade, velocidade e eficácia, especialmente em problemas que envolvem grandes volumes de dados ou características de alta dimensionalidade. Ele é comumente utilizado em aplicações como filtragem de spam, análise de sentimentos, diagnóstico médico e classificação de textos. Para muitos iniciantes no campo do aprendizado de máquina, o Naive Bayes representa uma introdução acessível e poderosa. Sua formulação matemática clara e intuitiva oferece um ponto de partida ideal para entender conceitos como probabilidade condicional e inferência estatística.

Embora tenha sido projetado com base em hipóteses simplificadoras, o Naive Bayes se destaca por sua robustez em diversos cenários, sendo frequentemente uma escolha inicial para resolver problemas de classificação antes de explorar métodos mais complexos.

E. Aplicações

A implementação do classificador Naive Bayes será conduzida utilizando a linguagem Python no ambiente Google Colaboratory, com o suporte de bibliotecas amplamente reconhecidas como Scikit-learn, Pandas, NumPy e SciPy. A Scikit-learn facilitará a aplicação do algoritmo Naive Bayes, enquanto Pandas e NumPy serão responsáveis pela manipulação e análise eficiente dos dados, e SciPy complementará as operações matemáticas e estatísticas. Essa combinação de ferramentas permite a exploração completa do classificador Naive Bayes no contexto de cibersegurança.

Os dados do projeto passarão por uma etapa essencial de análise e tratamento para garantir qualidade e relevância. Em seguida, serão divididos em dois subconjuntos: treinamento e teste. O conjunto de treinamento será usado para ajustar o modelo e identificar padrões, enquanto o conjunto de teste avaliará a eficiência do classificador, garantindo sua capacidade de generalização em novos dados. Essa abordagem permitirá medir métricas de desempenho, como precisão, recall e acurácia, essenciais para validar a eficácia do classificador em detectar comportamentos maliciosos no tráfego de rede. Dessa forma, o projeto oferecerá uma base sólida para a identificação de ataques cibernéticos e para a compreensão das características mais relevantes na classificação, contribuindo significativamente para o fortalecimento da cibersegurança.

V. Cronograma de Atividades

TABLE I
Cronograma de Atividades

Atividades		
Datas	***Ação programada***	***Detalhes***
10/12/2024	Formação de equipe	Em sala de aula
11/12/2024	Seleção do dataset	Em grupo
12/12 - 24/12/2024	Elaboração de ideias	Em grupo
25/12/2024	Início da escrita da proposta	Em grupo
24/01/2025	Finalização da proposta do projeto	Em grupo
27/01/2025	**Entrega da proposta**	**-**
03/02/2025	Divisão de tarefas	Em grupo
05/02 - 25/02/2025	Desenvolvimento do projeto	Colab/Python
03/03 - 09/03/2025	Elaboração do relatório	Em grupo/Individual
10/03/2025	Elaboração dos slides	Em grupo
15/03 - 25/03/2025	Período de correções e finalização	Em grupo/Individual
26/03/2025	**Entrega final do projeto**	**-**

References

[1] Dataset CIC-IDS2017, disponível em https://www.unb.ca/cic/datasets/ids-2017.html

[2] Classificador Naive Bayes, disponível em https://scikit-learn.org/stable/modules/naive_bayes.html