Aprendizado de Máquina Aplicado à Detecção de Ataques DDoS com Abordagens de Inteligência Artificial Explicável (XAI)

Everton Juan de Souza
Instituto Federal de Pernambuco
Palmares, Brasil
0009-0004-9428-4792

Adriano Henrique de Melo França
Instituto Federal de Pernambuco
Palmares, Brasil
0000-0003-3679-3519

Anderson Bispo dos Santos
Universidade de Pernambuco
Palmares, Brasil
0009-0007-9880-7986

Diogo Lopes da Silva
Instituto Federal de Pernambuco
Palmares, Brasil
0009-0004-7649-1448

***Abstract*—The accelerated growth of cyberattacks in recent years has posed a significant threat to the security of computer systems worldwide, especially due to their increasing complexity. Among these threats, Distributed Denial of Service (DDoS) attacks stand out for aiming to make services unavailable by overloading systems with malicious traffic. Given the limitations of traditional signature-based detection methods, this study aims to analyze the application of machine learning techniques for detecting DDoS attacks using a public dataset, as well as to employ explainable artificial intelligence approaches to make the results more transparent and interpretable. The methodology involved the use of the DDoS SDN dataset, with preprocessing steps that included removing null values and irrelevant columns, followed by data normalization using Standard Scaler and One Hot Encoder. The models applied were K-Nearest Neighbors (KNN), Decision Tree, and Random Forest, and were evaluated using metrics such as accuracy, precision, recall, and F1-Score. The results showed high performance from all models, with Decision Tree and Random Forest achieving 100% in all evaluated metrics, while KNN achieved 97.22% accuracy and a 96.43% F1-Score. These findings highlight the effectiveness of machine learning models in identifying malicious patterns and demonstrate their potential to enhance precision, adaptability, and interpretability in the detection of DDoS attacks.**

***Keywords*—Cyberattacks, DDoS, Machine Learning, Threat Detection, Explainable Artificial Intelligence.**

***Resumo*—O crescimento acelerado dos ataques cibernéticos nos últimos anos tem representado uma ameaça significativa à segurança de sistemas computacionais em todo o mundo, especialmente devido à sua complexidade crescente. Dentre esses ataques, destaca-se o Distributed Denial of Service (Negação de Serviço Distribuída - DDoS), que busca tornar serviços indisponíveis por meio da sobrecarga de tráfego malicioso. Frente às limitações de métodos tradicionais baseados em assinaturas, este trabalho tem como objetivo analisar a aplicação de técnicas de machine learning na detecção de ataques DDoS utilizando um conjunto de dados público, além de empregar abordagens de inteligência artificial explicável para tornar os resultados mais compreensíveis e transparentes. A metodologia empregou o conjunto de dados DDoS SDN dataset, passando por etapas de tratamento, exclusão de dados nulos e utilização de bibliotecas como Standard Scaler e One Hot Encoder. Os modelos utilizados foram K-Nearest Neighbors (KNN), Decision Tree e Random Forest, avaliados por métricas como acurácia, precisão, recall e F1-Score. Os resultados obtidos demonstraram alto desempenho dos modelos, com destaque para o Decision Tree e o Random Forest, que alcançaram 100% em todas as métricas avaliadas, enquanto o KNN obteve acurácia de 97,22% e F1-Score de 96,43%. Esses resultados evidenciam o potencial dessas técnicas na detecção eficaz de padrões maliciosos, contribuindo para soluções mais precisas, adaptáveis e explicáveis no combate a ataques DDoS.**

***Palavras-chave*—Ataques Cibernéticos, DDoS, Machine Learning, Detecção de Ameaças, Inteligência Artificial Explicável.**

I. Introdução

O crescimento da conectividade digital e a maior dependência de serviços online tornaram as infraestruturas de rede mais vulneráveis a ameaças cibernéticas. Entre essas ameaças, os ataques de negação de serviço distribuído (DDoS) têm se destacado pela sua capacidade de comprometer a disponibilidade de sistemas ao sobrecarregá-los com um alto volume de tráfego malicioso.

De acordo com a reportagem do site Tecnoblog [1], um ataque com pico de 73 Tbps foi registrado em 2023, sendo considerado o mais volumoso já registrado. Casos semelhantes também impactaram serviços governamentais brasileiros, como o Bolsa Família [2], e plataformas digitais como o Github [3],

evidenciando a gravidade desses ataques e demonstrando que nem mesmo as maiores empresas de tecnologia estão imunes.

Nesse cenário, a detecção precoce desses ataques tornou-se um desafio para a segurança da informação. De acordo com Marchi [4], técnicas de aprendizado de máquina podem ser aplicadas de forma eficiente na identificação de comportamentos anômalos em redes, permitindo uma análise mais dinâmica e inteligente dos dados.

A utilização de algoritmos de *Machine Learning* (ML) para a detecção de ataques DDoS tem mostrado resultados promissores, como discutido por Silva [5], que destaca a capacidade desses métodos em identificar padrões maliciosos e reagir a novas ameaças com maior precisão. Além disso, iniciativas voltadas à explicabilidade dos modelos de inteligência artificial, como as abordadas por Carvalho [6], contribuem para tornar os sistemas de defesa mais transparentes e confiáveis.

Neste contexto, este trabalho tem como objetivo analisar a aplicação de técnicas de *Machine Learning* na detecção de ataques DDoS em um conjunto de dados público e utilizar técnicas de inteligência artificial explicável para tornar os resultados mais compreensíveis e transparentes.

II. Referencial Teórico

Esta seção apresenta os principais conceitos necessários para uma melhor compreensão deste trabalho.

A. Ataque de Negação de Serviço Distribuído

Os ataques de negação de serviço distribuídos têm como objetivo tornar sites, aplicativos ou serviços online indisponíveis, sobrecarregando-os com grandes volumes de tráfego malicioso, como solicitações automatizadas e pacotes falsos.

De acordo com o trabalho de Oliveira [7], os ataques DDoS têm obtido bastante importância, principalmente devido à sofisticação e coordenação na forma em que estes ataques são executados, fazendo com que a prevenção e rastreamento tenham uma dificuldade elevada. Isso ocorre devido ao grande número de máquinas atacantes envolvidas e ao uso de técnicas para forjar endereços IP (IP Spoofing), que escondem a origem verdadeira dos pacotes.

No geral, ataques DDoS geram volumes de dados inesperados, chegando a Terabytes. Eles exploram os recursos disponíveis nos sistemas computacionais, como largura de banda e diversidade resultante da distribuição geográfica dos dispositivos, como mencionado por Pelloso [8]. Outro aspecto abordado neste trabalho é a abrangência das redes de dispositivos infectados (botnets), uma vez que podem ser compostas por dispositivos móveis com vulnerabilidades exploradas.

Segundo De Neira [9], a tarefa de identificar indícios da preparação dessas ameaças é desafiadora, pois os atacantes evitam conduzir ações que impactam nas características do tráfego de rede. Por causa disso, a preparação desses ataques é frequentemente confundida com o tráfego normal da rede, por produzir um volume de dados ínfimo comparado à investida maliciosa.

B. Machine Learning

Machine Learning (ML) é um ramo da Inteligência Artificial (IA) que permite que sistemas computacionais aprendam a realizar tarefas automaticamente, com base nos dados inseridos, sem serem explicitamente programados para isso.

De acordo com Mattos [10], o principal objetivo de um modelo de ML é construir um sistema de computador que aprenda com um banco de dados pré–definido e gere, ao final, um modelo de predição, classificação ou detecção. É citado que esses algoritmos estão difundidos em diversas áreas, como sistemas bancários para detecção de fraudes, segurança de dados, logística de empresas e entre outras.

Segundo De Souto [11], as técnicas de machine learning, em português, Aprendizado de Máquina (AM), podem ser divididas, de maneira geral, em aprendizado supervisionado e aprendizado não supervisionado. Se antes do processo de aprendizado, o modelo recebe um conjunto de exemplos, cada exemplo sendo formado por um conjunto de atributos de entrada e saída (rótulos), então esse tipo de aprendizado pode ser classificado como aprendizado supervisionado.

Já o aprendizado não supervisionado é realizado quando, para cada exemplo, apenas os atributos de entrada estão disponíveis. Essas técnicas de aprendizado são utilizadas quando o objetivo for encontrar em um conjunto de dados padrões ou tendências (aglomerados) que auxiliem o entendimento desses dados.

C. Inteligência Artificial Explicável

A Inteligência Artificial Explicável (Explainable Artificial Intelligence — XAI) é um conjunto de técnicas que busca tornar os modelos de IA mais compreensíveis e transparentes para os humanos, visto que os algoritmos tradicionais geralmente são vistos como “caixas-pretas” por sua complexidade e opacidade.

De acordo com Bavaresco [12], o termo caixa-preta refere-se a sistemas cujo funcionamento não é transparente ou facilmente compreensível aos seres humanos e que, embora esses modelos sejam capazes de processar grandes volumes de dados e identificar padrões complexos, sua estrutura dificulta a interpretação dos cálculos e das camadas de processamento que levam a uma determinada saída. O estudo de Tjoa [13] complementa que o objetivo central do XAI é justificar a confiabilidade das decisões automatizadas e fornecer explicações compreensíveis, especialmente em áreas sensíveis como a medicina, onde erros podem afetar vidas.

Segundo o estudo de Andrade [14], um sistema XAI deve estar apto a explicar, de maneira apropriada a um ser humano, a lógica interna de sua predição: o que foi feito, o que está fazendo agora e o que acontecerá a seguir.

O trabalho complementa que o nível de detalhes e as características da XAI devem ser estabelecidos levando-se em conta o público-alvo da explicação, trazendo como exemplo desenvolvedores de software que podem compreender pequenas redes bayesianas, mas que elas são um completo enigma para o usuário leigo. Da mesma maneira, explicações muito básicas podem ser insuficientes para que experts revisem ou auditem um algoritmo.

III. Metodologia

Inicialmente, foi realizada uma revisão bibliográfica sobre a aplicação de algoritmos de machine learning na detecção de ataques DDoS. Foram analisados artigos científicos e publicações relevantes para reconhecer métodos já empregados na detecção desse tipo de ameaça. Este estudo possibilitou entender as principais técnicas, restrições e obstáculos encontrados no campo. Em seguida foi escolhido o dataset e aplicado as técnicas de ML e XAI.

A. Escolha do Conjunto de Dados

Para realizar o treinamento dos modelos de machine learning, foi realizada uma pesquisa no Kaggle a fim de encontrar um conjunto de dados adequado para o problema em questão, alguns critérios foram utilizados para realizar essa seleção, como: ano de publicação e volume de dados. Além disso, o conjunto de dados precisava ter amostras de acesso normal e acesso malicioso, nesse caso, ataque DDoS. Por fim, o conjunto de dados escolhido foi o DDoS SDN dataset, ele possui 104.345 instâncias e 23 atributos e foi publicado no Kaggle no ano de 2021. O conjunto de dados em questão contém 3 atributos categóricos e 20 atributos numéricos (incluindo o rótulo) e possui tráfego benigno (0) e tráfego malicioso (1).

O dataset escolhido contém instâncias baseadas nos protocolos UDP, ICMP e TCP, indicando o uso de diferentes formas de ataques DDoS. No entanto, ele não realiza a classificação dos tipos específicos de ataque, limitando-se apenas a identificar se uma instância representa ou não um ataque DDoS. Tanto o título do conjunto de dados quanto a nomenclatura de alguns de seus atributos sugerem que ele foi gerado em um ambiente SDN (Software-Defined Network). Entretanto, o autor da base não disponibilizou informações detalhadas sobre o processo de coleta, a topologia da rede ou as ferramentas utilizadas. Diante disso, este trabalho adota a premissa de que os dados foram extraídos de uma rede definida por software, reconhecendo que esse tipo de ambiente apresenta características próprias de controle e gerenciamento de tráfego, como a separação entre o plano de controle e o plano de dados e o uso de controladores centralizados. Essas particularidades podem influenciar o comportamento dos fluxos de rede e, consequentemente, a distribuição das métricas registradas, aspecto que foi considerado na interpretação e análise dos resultados obtidos.

B. Tratamento dos Dados

Inicialmente, foi realizada a verificação da presença de valores ausentes no conjunto de dados. Constatou-se que dois atributos, `rx_kbps` e `tot_kbps`, apresentavam 506 valores nulos cada. A fim de avaliar a relevância desses dados ausentes, foi realizada uma análise dos valores existentes em cada coluna. Verificou-se que os dados faltantes correspondiam a menos de 1% do total, motivo pelo qual optou-se pela remoção de toda instância, uma vez que seu impacto na base era estatisticamente insignificante.

Na sequência, identificaram-se atributos considerados irrelevantes para o objetivo da análise, como o número do switch, os endereços IP de origem e destino, bem como as colunas referentes à duração em segundos e nanosegundos, essas últimas descartadas por já haver uma coluna representando o tempo total. Todas essas colunas foram, portanto, removidas do conjunto de dados.

Posteriormente, foi realizada a distinção entre variáveis numéricas e categóricas, permitindo a aplicação de técnicas apropriadas de pré-processamento. As variáveis numéricas foram normalizadas por meio do método *Standard Scaler*, enquanto as variáveis categóricas foram transformadas utilizando a técnica de *One Hot Encoder*.

Utilizou-se a biblioteca *Standard Scaler* para normalizar as variáveis numéricas, uma vez que ela padroniza os dados com média zero e desvio padrão um, sendo adequado para algoritmos sensíveis à escala, como por exemplo, o KNN. Já para as variáveis categóricas, aplicou-se a biblioteca *One Hot Encoder*, que transforma categorias em variáveis binárias, evitando atribuição de ordens indevidas.

Optou-se por essas técnicas por sua eficácia e compatibilidade com a maioria dos modelos de machine learning, em detrimento de métodos como *Min-Max Scaling* e *Label Encoding*, que não seriam ideais para o contexto analisado.

C. Seleção dos Modelos de ML

Com base nos estudos realizados e das características dos dados coletados, foram escolhidos 3 modelos de machine learning que mostraram-se adequados para o problema, são eles: KNN, Decision Tree e Random Forest.

O K-Nearest Neighbors (KNN) é um modelo baseado em instâncias que classifica uma amostra com base nos ”K”vizinhos mais próximos dela no espaço de atributos. Ele calcula a distância entre a nova amostra e os dados de treino

e decide a classe com base na maioria dos vizinhos. Sua simplicidade e boa performance o permite distinguir padrões com base na similaridade das amostras, sendo eficaz para problemas binários.

A árvore de decisão (Decision Tree) é um modelo que cria uma estrutura em forma de árvore, onde cada nó representa uma decisão com base em um atributo e as folhas representam a classe final. Ele divide os dados em subconjuntos de forma recursiva, buscando maximizar a separação entre classes. Por esse motivo, torna-se ideal para problemas de classificação binária, pois lida bem com dados categóricos e contínuos, fornecendo uma maior interpretabilidade, pois revela quais atributos são mais relevantes para a decisão de identificar um ataque.

O Random Forest é um modelo baseado em um conjunto de várias árvores de decisão, onde cada árvore é treinada com uma amostra aleatória dos dados. Ele cria várias árvores independentes e combina os resultados delas para tomar uma decisão mais precisa e estável. Por ser um modelo robusto contra *overfitting* e mais preciso do que uma árvore isolada, torna-se excelente para tarefas como detecção de ataques DDoS, pois combina múltiplas decisões para aumentar a acurácia e a generalização do modelo frente à variabilidade dos dados.

D. Treinamento dos Modelos

Para o treinamento dos modelos, os dados foram divididos em 70% para treino e 30% para teste, utilizando técnicas como o *random state* para garantir que a divisão dos dados seja sempre a mesma caso o código seja executado várias vezes (*reprodutibilidade*) e *stratify* para garantir que a proporção da classe alvo seja mantida tanto no treino, quanto no teste.

Após isso, foi utilizado um pipeline com os 3 modelos escolhidos, onde foram definidas algumas configurações para o treinamento dos mesmos. Para o KNN, foi definida uma quantidade de 7 vizinhos mais próximos e para o Decision Tree e Random Forest, foi utilizado o parâmetro *random state* com um valor fixo de 42 para controlar a aleatoriedade durante o processo de construção desses modelos. Além disso, foi definido o valor de 100 árvores para treinamento do Random Forest.

E. Comparação dos Modelos

Para comparar e avaliar o desempenho dos modelos escolhidos, foram utilizadas 4 métricas: acurácia, precisão, sensibilidade ou *recall* e F1-Score.

Acurácia (equação 1): Mede a proporção de previsões corretas (positivas e negativas) em relação ao total de amostras.

$$\text{Acurácia} = \frac{TP + TN}{TP + TN + FP + FN} \tag{1}$$

Precisão (equação 2): Indica a proporção de acertos entre as previsões positivas feitas pelo modelo.

$$\text{Precisão} = \frac{TP}{TP + FP} \tag{2}$$

Recall (equação 3): Mede a proporção de positivos reais que foram corretamente identificados pelo modelo.

$$\text{Recall} = \frac{TP}{TP + FN} \tag{3}$$

F1-Score (equação 4): É a média harmônica entre precisão e a sensibilidade, equilibrando as duas métricas, especialmente útil em dados desbalanceados.

$$F1 = \frac{2 \cdot \text{Precisão} \cdot \text{Recall}}{\text{Precisão} + \text{Recall}} \tag{4}$$

IV. Resultados

Os modelos escolhidos obtiveram bons resultados, visto que, são algoritmos adequados para problemas de classificação binária.

	Model	Accuracy	Precision	Recall	F1-Score
0	KNN	0.972233	0.966441	0.962225	0.964328
1	Decision Tree	1.000000	1.000000	1.000000	1.000000
2	Random Forest	1.000000	1.000000	1.000000	1.000000

Tabela I
Resultado do Conjunto de Treino.

Os dados relacionados ao tráfego de rede costumam ser bem estabelecidos, tendo pouca ou nenhuma variação dependendo daquilo que esteja sendo monitorado, facilitando a identificação de padrões pelos algoritmos de classificação mais robustos, como os utilizados neste trabalho.

No conjunto de treino, os modelos selecionados obtiveram os resultados apresentados na Tabela I, onde o *Decision Tree* e *Random Forest* alcançaram 100% em todas as métricas estabelecidas, já o *KNN* alcançou um resultado inferior aos demais, tendo 97% de acurácia e 96% nas outras métricas.

Já no conjunto de teste, onde o objetivo era prever e classificar o tráfego da rede em “Normal” ou “Anomaly”, os modelos obtiveram os seguintes resultados:

A Tabela II apresenta uma pequena amostra dos resultados obtidos no conjunto de teste. Como neste conjunto o objetivo é classificar os dados em *Normal* ou *Anomaly*, não há a presença de rótulos, ou seja, não existe uma coluna que indique a classe real de cada acesso. Dessa forma, os modelos precisam realizar

	KNN	Decision Tree	Random Forest
0	Normal	Normal	Normal
1	Anomaly	Anomaly	Anomaly
2	Normal	Normal	Normal
3	Normal	Normal	Normal
4	Normal	Normal	Normal

Tabela II
RESULTADO DO CONJUNTO DE TESTE.

a predição com base no conhecimento adquirido durante a etapa de treinamento.

Observa-se que, na amostra exibida, todos os algoritmos analisados produziram os mesmos resultados, o que sugere que, mesmo apresentando desempenho inferior aos demais no conjunto de treino, o *KNN* demonstrou-se eficaz na classificação dos acessos.

A. *Avaliação de Desempenho*

Para validar o desempenho dos algoritmos selecionados, foram utilizadas algumas métricas de avaliação em forma gráfica, como a matriz de confusão e a curva ROC.

A matriz de confusão é um gráfico que resume o desempenho de um modelo de classificação, apresentando a quantidade de acertos e erros em cada classe por meio dos valores de verdadeiros positivos, falsos positivos, verdadeiros negativos e falsos negativos.

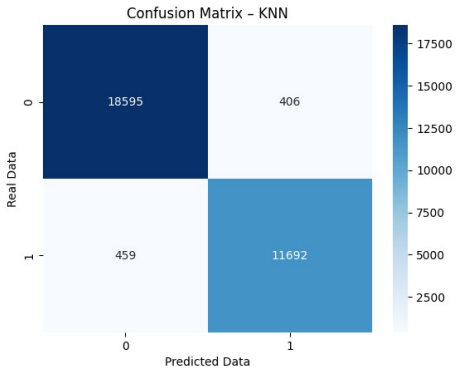

Figura 1. Matriz de Confusão - KNN.

Como mostrado na Figura 1, o KNN teve uma boa performance, embora não perfeita como os outros modelos. Ele errou algumas amostras das duas classes e isso pode ter ocorrido pelo fato de que o KNN é sensível à escala dos dados, valores ruidosos e densidade no espaço de vizinhança.

Porém, a performance ainda é bastante alta, o que mostra que o modelo aprendeu bem, mas lida com mais ruído ou sobreposição nas classes do que os outros. Ele conseguiu identificar 18.595 verdadeiros negativos, 11.692 verdadeiros positivos, 406 falsos positivos e 459 falsos negativos.

Na Figura 2 é mostrado o desempenho da Árvore de Decisão, que apresentou uma classificação perfeita, o que é intrigante, visto que o *Decision Tree* isoladamente tende a ter menor capacidade de generalização comparado a modelos como o *Random Forest*. Os bons resultados podem ter acontecido pelo fato de que os padrões nos dados estavam muito bem definidos, facilitando a separação por regras. Além disso, pode indicar *overfitting*, já que árvores muito profundas podem memorizar os dados. Dessa forma, ele conseguiu identificar 19.001 verdadeiros negativos, 12.151 verdadeiros positivos, 0 falsos positivos e 0 falsos negativos.

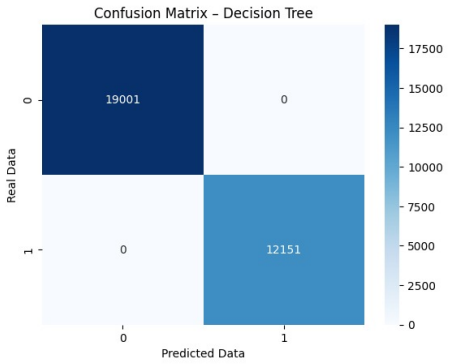

Figura 2. Matriz de Confusão - Decision Tree.

Na Figura 3, nota-se que o modelo Random Forest também teve um desempenho perfeito nesse conjunto de dados.

Ele classificou corretamente todas as amostras, sem cometer nenhum erro. Isso pode indicar que o modelo generalizou muito bem os padrões do dataset ou ter havido balanceamento ideal das classes e/ou boas features no processo de treinamento. Sendo assim, ele conseguiu identificar 19001 verdadeiros negativos, 12151 verdadeiros positivos, 0 falsos positivos e 0 falsos negativos.

Já a curva ROC (Receiver Operating Characteristic) representa graficamente a relação entre a taxa de verdadeiros positivos (sensibilidade) e a taxa de falsos positivos, permitindo avaliar o desempenho do modelo em diferentes limiares de classificação e sua capacidade de discriminar entre as classes. Quanto mais a curva estiver próxima do canto superior esquerdo, melhor o desempenho do modelo.

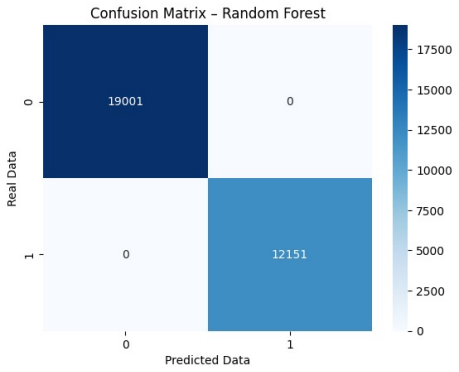

Figura 3. Matriz de Confusão - Random Forest.

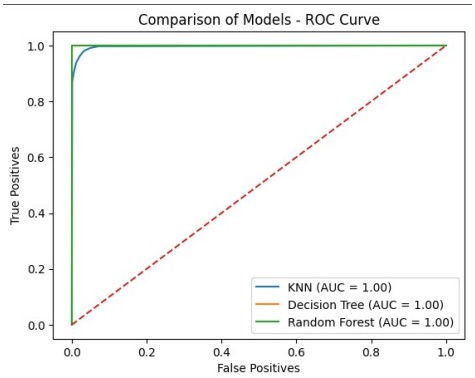

Figura 4. Curva ROC.

Na Figura 4, percebe-se que a área sob a curva (AUC) quantifica essa performance da seguinte maneira:

O *KNN* alcançou uma curva perfeita, atingindo 100%. Isso mostra que, apesar de pequenos erros na matriz de confusão, o *KNN* ainda tem grande capacidade de discriminar entre as classes, mesmo com alguns erros pontuais.

Assim como o *KNN*, o *Decision Tree* alcançou AUC perfeita. Isso significa que a árvore conseguiu separar as classes de forma exata em todos os limiares. Esse desempenho é surpreendente para uma árvore simples, podendo indicar que o *dataset* esteja muito "limpo" ou bem estruturado, fazendo com que o modelo tenha se ajustado demais aos dados.

O *Random Forest* também obteve uma curva perfeita, atingindo 100%. Isso mostra que o modelo teve desempenho ideal, confirmando o que já foi observado na matriz de confusão, onde o mesmo classificou todas as amostras corretamente. Esse resultado pode indicar dados muito bem separados, mas também levantar suspeitas de *overfitting*.

Essas métricas foram escolhidas por oferecerem uma avaliação mais completa do que as métricas utilizadas anteriormente, como a acurácia, que pode ser enganosa em cenários com distribuição desigual entre as classes. Com a matriz de confusão e a curva ROC, é possível demonstrar os resultados de forma gráfica, permitindo uma melhor compreensão dos dados e uma tomada de decisão mais embasada.

B. Avaliação do Resultado

Para verificar a integridade dos resultados obtidos e sanar algumas hipóteses, foi necessário visualizar algumas informações relevantes, como: a raiz da árvore de decisão e um gráfico de dispersão contendo as duas variáveis com maior correlação em relação ao alvo.

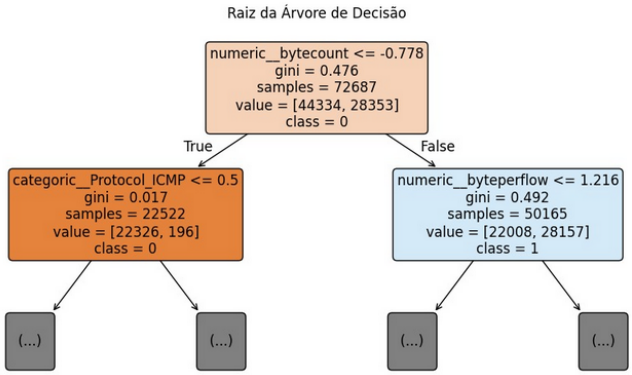

Figura 5. Raiz da Árvore de Decisão.

A raiz da árvore de decisão permite visualizar as primeiras divisões realizadas pelo modelo. Essa visualização permite compreender quais atributos foram considerados mais relevantes nas decisões iniciais, auxiliando na interpretação do funcionamento do modelo e na identificação de possíveis padrões nos dados (Figura 5).

Observa-se que o primeiro critério de divisão envolve o atributo "bytecount", seguido por "Protocol_ICMP" e "byteperflow", indicando a importância desses atributos na separação entre as classes. Além disso, existem informações importantes, como o *gini*, que indica o índice de impureza das amostras, medindo o quanto os dados estão misturados entre as classes, quanto mais próximo de 0, mais "puro" é o grupo. O *samples* mostra o total de amostras que chegaram até esse nó e o *value* indica quantas amostras são de cada classe.

O gráfico de dispersão entre os atributos "pktcount" e "bytecount", segmentado por classe, foi utilizado para explorar visualmente a distribuição dos dados.

A Figura 6 revela uma separação perceptível entre as classes, com duas tendências bem definidas. Esse tipo de visualização é útil para detectar agrupamentos ou padrões que podem contribuir para a eficácia de modelos de classificação, além

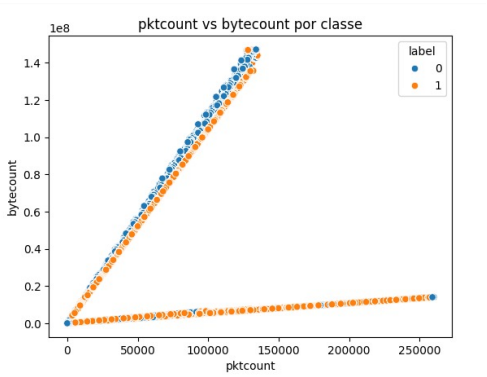

Figura 6. Gráfico de Dispersão.

de reforçar a relevância dessas variáveis no processo de aprendizado.

V. Aplicação de IA Explicável

Para implementação de IA Explicável no trabalho em questão, foram utilizadas duas técnicas: LIME e SHAP. O LIME foi utilizado para o modelo KNN e SHAP para o modelo Random Forest, pois esses modelos são pouco transparentes em como chegaram nos resultados obtidos. Para o Decision Tree, foi utilizado apenas a impressão da raiz da árvore, como demonstrado na Figura 5, visto que através dela é possível obter informações relevantes de como o modelo chegou no resultado.

A. LIME

O *Local Interpretable Model-agnostic Explanations* (LIME) é um método de explicabilidade que visa facilitar a compreensão do funcionamento de modelos preditivos complexos, fornecendo explicações para cada previsão feita. Ele realiza essa tarefa ao criar, de maneira local e específica, um modelo interpretável, normalmente linear ou fundamentado em regras simples, que se alinha ao comportamento do modelo original na vizinhança da instância em análise.

A ideia central é que, mesmo que o modelo global seja muito complexo para ser interpretado diretamente, é possível obter explicações confiáveis em nível local, ou seja, explicações sobre por que o modelo tomou determinada decisão para uma entrada específica.

Na Figura 7, observa-se que o modelo classificou a classe "Normal" com 86% de confiança e "Anomaly" com 14%. Na parte esquerda, temos as *features* que contribuíram para a predição "Normal" (em azul) e na direita, as *features* que puxaram a predição para "Anomaly" (em laranja).

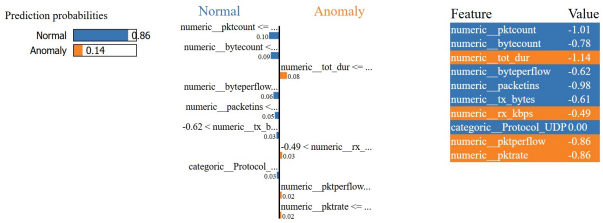

Figura 7. Aplicação do LIME no KNN.

B. SHAP

O *SHapley Additive exPlanations* (SHAP) é um método de explicabilidade fundamentado na teoria dos valores de Shapley, que é um princípio da teoria dos jogos cooperativos. O SHAP visa fornecer a cada recurso uma contribuição justa e consistente para a previsão realizada por um modelo.

A técnica avalia o quanto cada variável contribuiu, de forma positiva ou negativa, para a previsão de uma instância, levando em conta todas as combinações de entrada possíveis. Independentemente do modelo empregado, o resultado é uma explicação coerente, justa em termos globais e precisa em termos locais, mesmo que haja implementações otimizadas para modelos como árvores de decisão.

A Figura 8 apresenta um *SHAP Interaction Plot*, utilizado para mostrar interações entre variáveis, ou seja, como duas *features* combinadas influenciam a predição de um modelo. O eixo Y representa as variáveis analisadas: `numeric_pktcount` e `numeric_dt`. Já o eixo X mostra os valores de interação SHAP: quanto a combinação entre duas variáveis influencia a predição. Cada ponto representa uma instância do conjunto de dados e as cores representam os valores das *features* envolvidas na interação: rosa indica valores altos da variável e azul valores baixos.

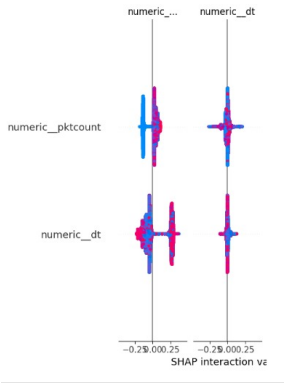

Figura 8. Aplicação do SHAP no Random Forest.

A diagonal principal (*numeric_dt* com *numeric_dt*) mostra

o efeito isolado da feature sobre a predição. Já os elementos fora da diagonal (*numeric_pktcount* × *numeric_dt*) mostram o efeito conjunto entre duas variáveis, ou seja, como a influência de uma variável muda dependendo do valor da outra.

VI. Conclusão e Trabalhos Futuros

Este artigo apresentou o monitoramento de ataques DDoS em uma base de dados pública, utilizando algoritmos de *Machine Learning*, como KNN, Decision Tree e Random Forest, além de técnicas de inteligência artificial explicável (*XAI*) para maior transparência dos resultados.

Inicialmente, os dados passaram por uma etapa de tratamento. Após esta etapa, foram separados em 70% para treino e 30% para teste. Na fase de treino, os algoritmos escolhidos obtiveram uma taxa de acurácia superior a 97%, com destaque para o Decision Tree e o Random Forest, que alcançaram 100% em todas as métricas avaliadas. Na fase de teste, os modelos obtiveram resultados semelhantes, sendo avaliados por métricas adicionais, como a matriz de confusão e a curva ROC.

Para melhor compreensão desses resultados, foram utilizadas técnicas para imprimir a raiz da árvore de decisão, além de métodos de XAI, como LIME e SHAP, aplicados ao KNN e ao Random Forest, respectivamente. Os resultados mostraram-se eficazes, visto que os modelos de ML escolhidos apresentaram bom desempenho na classificação de tráfegos normais e maliciosos, e as técnicas de XAI possibilitaram uma visualização gráfica do processo de decisão dos modelos.

Como trabalhos futuros, propõe-se o treinamento dos modelos selecionados em um novo conjunto de dados contendo diversos tipos de ataques de rede, a fim de validar o comportamento dos modelos em cenários de tráfego mais robustos. Além disso, sugere-se a comparação do desempenho desses modelos em uma base de dados real, gerada a partir de ferramentas de monitoramento, como o *Suricata* ou o *TCPDump*.

Referências

[1] Tecnoblog. (2023, jun) 73 tb/s: maior ataque ddos da história atinge cliente da cloudflare. Acesso em: 15 jul. 2025. [Online]. Available: https://tecnoblog.net/noticias/73-tb-s-maior-ataque-ddos-da-historia-atinge-cliente-da-cloudflare/

[2] Veja. (2023, ago) Ddos: o ataque hacker que derruba redes oficiais e já mirou bolsa família. Acesso em: 15 jul. 2025. [Online]. Available: https://veja.abril.com.br/coluna/maquiavel/ddos-o-ataque-hacker-que-derruba-redes-oficiais-e-ja-mirou-bolsa-familia/

[3] UFRJ. (2018, mar) Github passou pelo maior ataque ddos já registrado. Acesso em: 15 jul. 2025. [Online]. Available: https://seguranca.tic.ufrj.br/alertas/github-passou-pelo-maior-ataque-ddos-ja-registrado/

[4] A. José Marchi, M. Zazeri Fonseca, and J. Bodê, "Machine learning: Aplicabilidade em monitoramento de redes," in *FatecSeg - Congresso De Segurança Da Informação*, 2023, acessado em: 07 out. 2025. [Online]. Available: https://fatecseg.fatecourinhos.edu.br/index.php/fatecseg/article/view/120

[5] R. R. Silva *et al.*, "Detecção de ataques de negação de serviço distribuídos com algoritmos de aprendizado de máquina," in *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*. SBC, 2024, pp. 226–241.

[6] A. B. Carvalho *et al.*, "Abrindo a caixa-preta – aplicando ia explicável para aprimorar a detecção de sequestros de prefixo," in *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*. SBC, 2024, pp. 16–31.

[7] L. Oliveira *et al.*, "Avaliação de proteção contra ataques de negação de serviço distribuídos (ddos) utilizando lista de ips confiáveis," in *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*. SBC, 2007, pp. 177–190.

[8] M. Pelloso *et al.*, "Um sistema autoadaptável para predição de ataques ddos fundado na teoria da metaestabilidade," in *Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos (SBRC)*. SBC, 2018, pp. 726–739.

[9] A. B. D. Neira *et al.*, "Engenharia de sinais precoces de alerta para a predição de ataques ddos," in *Workshop de Gerência e Operação de Redes e Serviços (WGRS)*. SBC, 2023, pp. 139–152.

[10] G. M. de Mattos Paixão *et al.*, "Machine learning na medicina: revisão e aplicabilidade," *Arquivos Brasileiros de Cardiologia*, vol. 118, no. 1, pp. 95–102, 2022.

[11] M. C. P. D. Souto *et al.*, "Técnicas de aprendizado de máquina para problemas de biologia molecular," *Sociedade Brasileira de Computação*, vol. 1, no. 2, 2003.

[12] M. Z. Bavaresco and C. G. Webber, "Ia explicável para reduzir a assimetria de informação no consumo: uma análise comparativa de ferramentas e implicações educacionais," *Redin-Revista Educacional Interdisciplinar*, vol. 13, no. 2, pp. 59–77, 2024.

[13] E. Tjoa and C. Guan, "A survey on explainable artificial intelligence (xai): Toward medical xai," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 11, pp. 4793–4813, 2020.

[14] O. M. D. Andrade, "Da "caixa-preta" à "caixa de vidro": o uso da explainable artificial intelligence (xai) para reduzir a opacidade e enfrentar o enviesamento em modelos algorítmicos," *Direito Público*, 2022.