APRENDIZADO DE MÁQUINA APLICADO À DETECÇÃO DE ATAQUES DDOS COM ABORDAGENS DE INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL (XAI)

Everton Juan de Souza

ejs34@discente.ifpe.edu.br

Orientador: Adriano Henrique de Melo França

adriano.franca@palmares.ifpe.edu.br

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-chaves: Ataques Cibernéticos; DDoS; Machine Learning; Detecção de Ameaças; Inteligência Artificial Explicável.

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

Instituto Federal de Pernambuco. *Campus* Palmares. Curso de Análise e Desenvolvimento de Sistemas.

30 de julho de 2025.

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.

1 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 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 reportagem do Tecnoblog (TECNOBLOG, 2023), 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 (VEJA, 2023), e plataformas digitais como o *GitHub* (UFRJ, 2018), 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. Marchi e Fonseca (2023) destacam que 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), em português, Aprendizado de Máquina (AM) para a detecção de ataques DDoS têm mostrado resultados promissores, como discutido por Silva et al. (2024), que ressaltam a capacidade desses métodos em identificar padrões maliciosos e reagir a novas ameaças com mais precisão. Além disso, iniciativas voltadas à explicabilidade dos modelos de inteligência artificial, como as abordadas por Carvalho et al. (2024), 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.

2 REFERENCIAL TEÓRICO

2.1 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 Oliveira et al. (2007), 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 o rastreamento apresentem 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 et al. (2018). 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 et al. (2023), 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.

2.2 Machine Learning

Machine Learning 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 Paixão et al. (2022), 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. Esses algoritmos estão difundidos em diversas áreas, como sistemas bancários para detecção de fraudes, segurança de dados e logística de empresas, entre outros.

Segundo De Souto et al. (2003), as técnicas de *machine learning* 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 um formado por atributos de entrada e saída (rótulos), então esse tipo de aprendizado pode ser classificado como aprendizado supervisionado.

Já o aprendizado não supervisionado ocorre quando, para cada exemplo, apenas os atributos de entrada estão disponíveis. Essas técnicas de aprendizado são utilizadas quando o objetivo é identificar, em um conjunto de dados, padrões ou tendências (aglomerados) que auxiliem no entendimento das informações.

2.3 Inteligência Artificial Explicável

A Inteligência Artificial Explicável (*Explainable Artificial Intelligence* – XAI) corresponde a um conjunto de técnicas que buscam tornar os modelos de IA mais compreensíveis e transparentes para os seres humanos, uma vez que algoritmos tradicionais são frequentemente tratados como “caixas-pretas”, em razão de sua complexidade e opacidade.

De acordo com Bavaresco (2012), o termo caixa-preta refere-se a sistemas cujo funcionamento não é transparente ou facilmente compreensível, e embora tais 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 conduzem a determinada saída.

Segundo Andrade (2013), um sistema de XAI deve ser capaz de explicar, de forma apropriada ao ser humano, a lógica interna de sua predição: o que foi realizado, o que está sendo processado e o que ocorrerá em seguida.

Esse autor ainda complementa que o nível de detalhamento e as características da XAI devem ser definidos considerando o público-alvo da explicação. Como exemplo, desenvolvedores de software podem compreender redes bayesianas simplificadas, enquanto tais modelos permanecem enigmáticos para usuários leigos. Da mesma forma, explicações excessivamente básicas mostram-se insuficientes para que especialistas revisem ou auditem um algoritmo.

3 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.

3.1 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*[1], 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).

[1] https://www.kaggle.com/datasets/aikenkazin/ddos-sdn-dataset

3.2 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*[2], enquanto as variáveis categóricas foram transformadas utilizando a técnica de *One Hot Encoder*[3].

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.

3.3 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[4], *Decision Tree*[5] e *Random Forest*[6].

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

[2] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.StandardScaler.html
[3] https://scikit-learn.org/stable/modules/generated/sklearn.preprocessing.OneHotEncoder.html
[4] https://scikit-learn.org/stable/modules/generated/sklearn.neighbors.KNeighborsClassifier.html
[5] https://scikit-learn.org/stable/modules/tree.html
[6] https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.RandomForestClassifier.html

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.

3.4 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*.

3.5 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.

$$(1) \qquad Acurácia = \frac{TP + TN}{TP+TN+FP+FN}$$

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

$$(2) \qquad Precisão = \frac{TP}{TP+FP}$$

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

$$(3) \qquad Recall = \frac{TP}{TP+FN}$$

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

$$F1 = 2.\frac{Precisão\ .\ Recall}{Precisão + Recall} \quad (4)$$

4 RESULTADO

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

Figura 1 - Resultado do Conjunto de Treino.

	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

Fonte: Próprio Autor.

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 Figura 1, 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:

Figura 2 - Resultado do Conjunto de Teste.

	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

Fonte: Próprio Autor.

A Figura 2 exibe uma pequena amostra dos resultados no conjunto de teste, como nesse conjunto o objetivo é classificar os dados em normais ou anômalos, ele não possui rótulos, ou seja, não possui a coluna que define o que cada acesso realmente é. Dessa forma, os modelos precisam realizar uma predição dos dados com base naquilo que foi aprendido na etapa de treinamento.

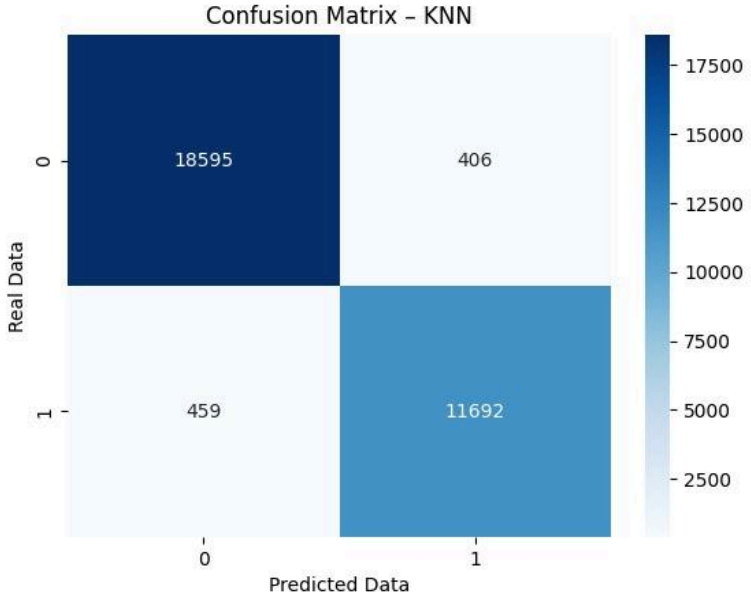
Nota-se que na amostra exibida, os algoritmos em questão tiveram os mesmos resultados, sugerindo que mesmo com um desempenho inferior aos demais na etapa de treino, o KNN mostra-se eficaz na classificação desses acessos.

4.1 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 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 3 - Matriz de Confusão - KNN.



Fonte: Próprio Autor.

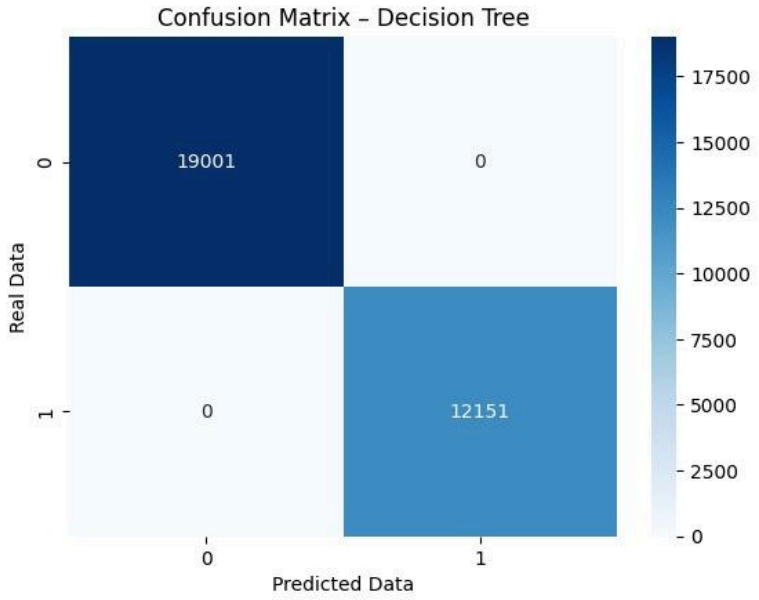
Como mostrado na Figura 3, o KNN teve uma boa performance, embora não perfeita como os outros modelos. Ele errou algumas amostras das duas classes e isso pode ter ocorrido pelo fato que o KNN é sensível à escala dos dados, valores ruidosos e densidade no espaço de vizinhança.

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 18595 verdadeiros negativos, 11692 verdadeiros positivos, 406 falsos positivos e 459 falsos negativos.

Na Figura 4 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 *Random Forest*. Os bons resultados podem ter acontecido pelo fato 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 19001 verdadeiros negativos, 12151 verdadeiros positivos, 0 falsos positivos e 0 falsos negativos.

Figura 4 - Matriz de Confusão - *Decision Tree*.

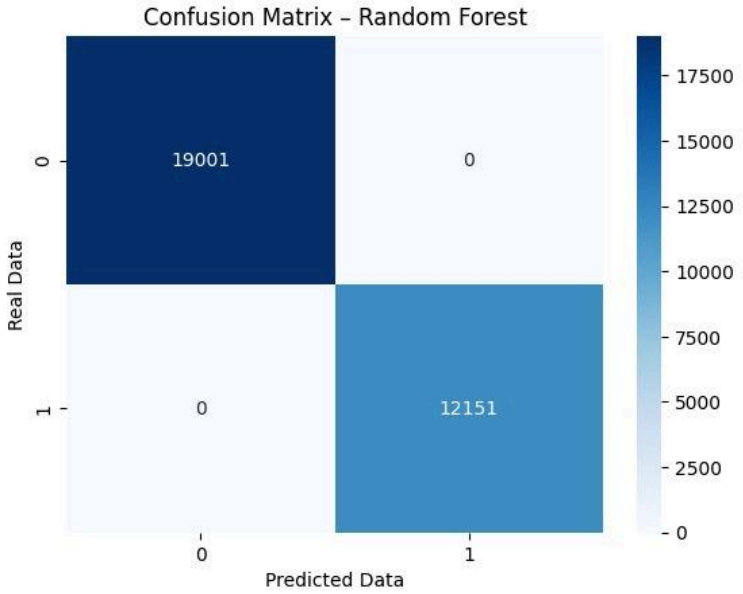


Fonte: Próprio Autor

Na Figura 5, 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.

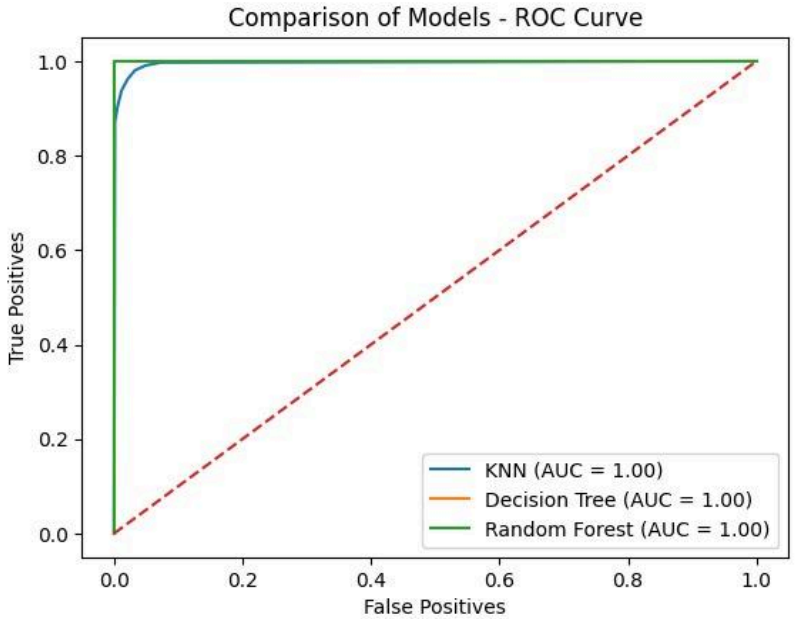
Figura 5 - Matriz de Confusão - *Random Forest*.



Fonte: Próprio Autor.

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 6 - Curva ROC



Fonte: Próprio Autor.

Na Figura 6, 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 e o modelo tenha se ajustado demais aos dados.

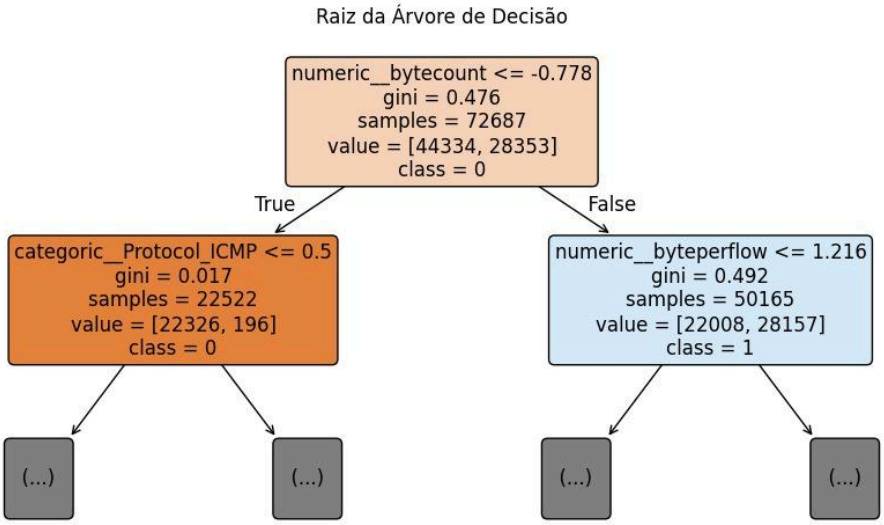
O *Random Forest* também obteve uma curva perfeita, atingindo 100%. Isso mostra que o modelo teve desempenho ideal, confirmando o que já foi observado na matriz de confusão, onde o mesmo classificou todas as amostras corretamente. Isso pode indicar dados muito bem separados, mas também pode levantar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

4.2 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 7 - Raiz da Árvore de Decisão.



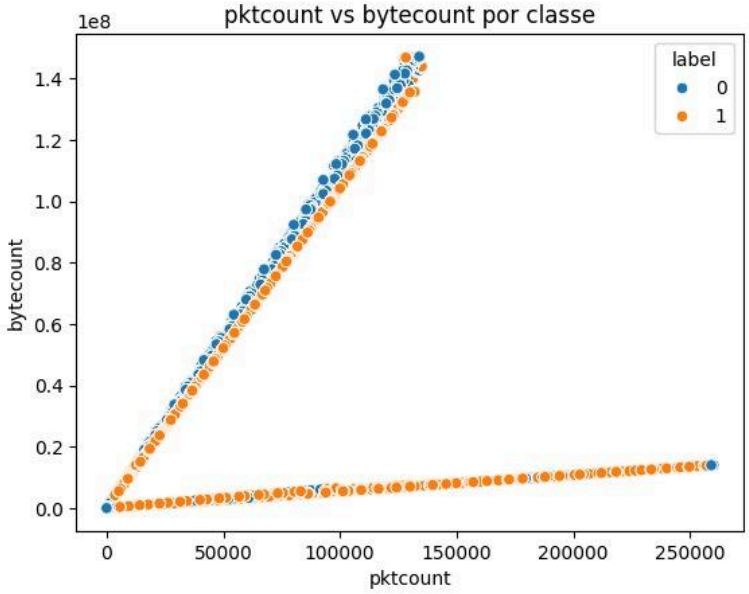
Fonte: Próprio Autor.

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 7).

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" o grupo. O samples que mostra o total de amostras que chegaram até esse nó e o value que mostra 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.

Figura 8 - Gráfico de Dispersão.



Fonte: Próprio Autor.

A Figura 8 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 de reforçar a relevância dessas variáveis no processo de aprendizado.

5 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 7, visto que através dela, é possível obter informações relevantes de como o modelo chegou no resultado.

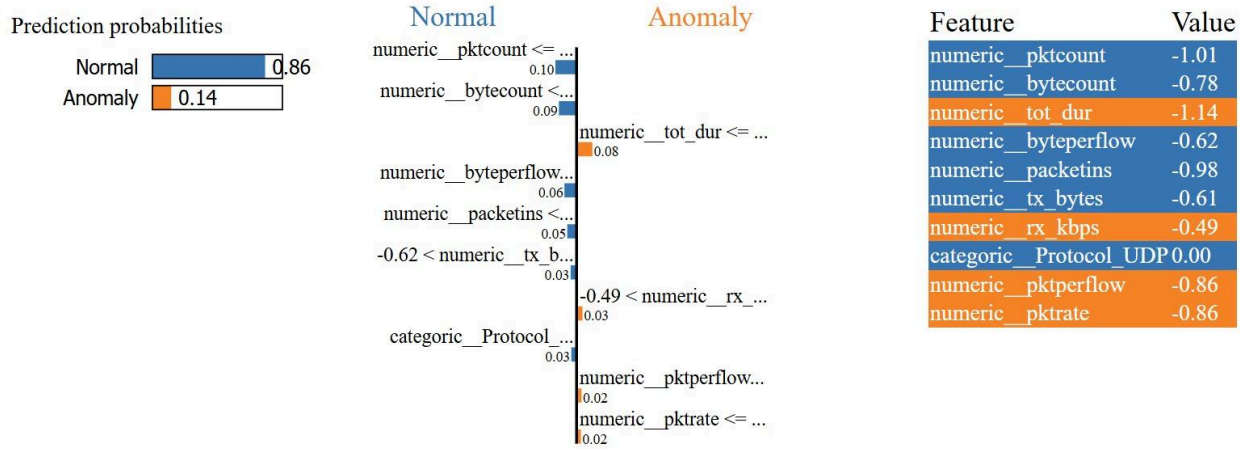
5.1 LIME

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 9, 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 9 - Aplicação do LIME no KNN.



Fonte: Próprio Autor.

5.2 SHAP

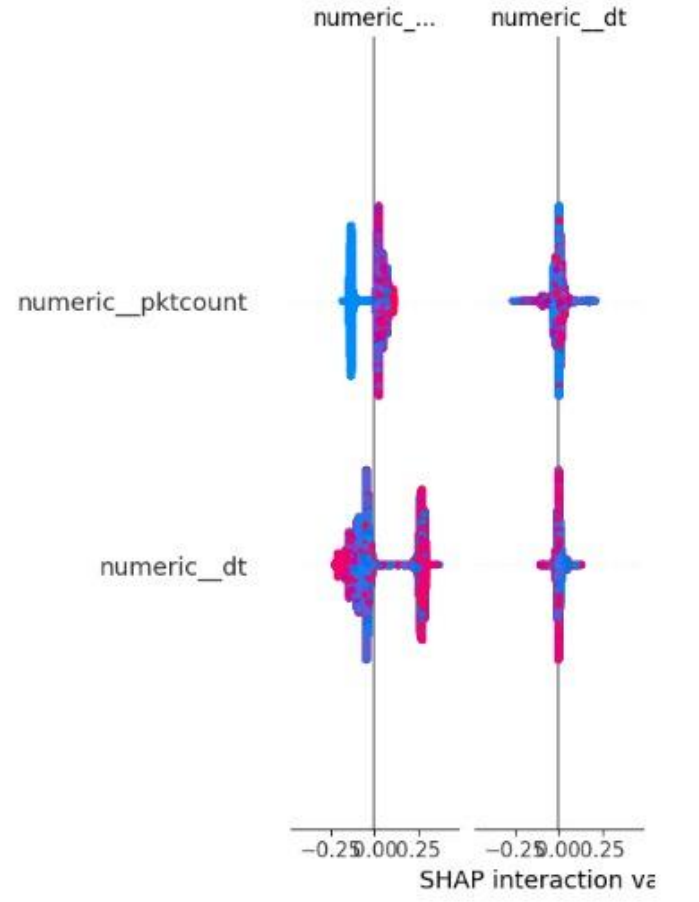
O *SHapley Additive exPlanations* (SHAP) é uma método de explicabilidade fundamentada na teoria dos valores de Shapley, que é um princípio da teoria dos jogos cooperativos. O SHAP visa fornecer a cada recurso uma contribuição justa e consistente para a previsão realizada por um modelo.

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 10 é 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 representa valores altos da variável e azul valores baixos.

Figura 10 - Aplicação do SHAP no *Random Forest.*



Fonte: Próprio Autor.

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.

6 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 utilizar técnicas de inteligência artificial explicável para maior transparência dos resultados. Inicialmente, esses dados passaram por uma etapa de tratamento. Após esta etapa, os dados foram separados em 70% para treino e 30% para teste, onde na fase de treino, os algoritmos escolhidos tiverem uma taxa de acurácia maior que 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 novas métricas, como matriz de confusão e curva ROC. Para melhor compreensão desses resultados, foram utilizadas técnicas para imprimir a raiz da árvore de decisão e técnicas de IA Explicável como LIME e SHAP para o KNN e *Random Forest* respectivamente.

Os resultados obtidos mostraram-se eficazes, visto que, os modelos de ML escolhidos tiveram um bom desempenho em classificar tráfegos normais e maliciosos e as técnicas de XAI utilizadas mostraram de forma gráfica como os modelos chegaram nos resultados.

Como trabalhos futuros, propõe-se o treinamento dos modelos selecionados em um novo conjunto de dados contendo diversos ataques de rede, a fim de validar como esses modelos se comportam no monitoramento de um tráfego mais robusto. Além disso, sugere-se a comparação do desempenho desses modelos em uma base de dados real, ou seja, uma base de dados gerada a partir de alguma ferramenta de monitoramento, como *Suricate* ou TCPDump.

REFERÊNCIAS

ANDRADE, Otávio Morato de. 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.

BAVARESCO, Maurício Zalamena; WEBBER, Carine Geltrudes. IA explicável para reduzir a assimetria de informação no consumo: uma análise comparativa de ferramentas e implicações educacionais. Redin-Revista Educacional Interdisciplinar, v. 13, n. 2, p. 59-77, 2024.

CARVALHO, Adriano B. 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. p. 16-31.

DE NEIRA, Anderson B. 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. p. 139-152.

DE SOUTO, M. C. P. et al. Técnicas de aprendizado de máquina para problemas de biologia molecular. Sociedade Brasileira de Computação, v. 1, n. 2, 2003.
MARCHI, Amadeu José; FONSECA, Maurício Zazeri. Machine learning: aplicabilidade em monitoramento de redes. 2023.

OLIVEIRA, Luis 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. p. 177-190.

PAIXÃO, Gabriela Miana de Mattos et al. Machine learning na medicina: revisão e aplicabilidade. Arquivos Brasileiros de Cardiologia, v. 118, n. 1, p. 95-102, 2022.
PELLOSO, Mateus 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. p. 726-739.
SILVA, Rodrigo R. 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. p. 226-241.

TECNOBLOG. 73 Tb/s: maior ataque DDoS da história atinge cliente da Cloudflare. Tecnoblog, 15 jun. 2023. Disponível em: https://tecnoblog.net/noticias/73-tb-s-maior-ataque-ddos-da-historia-atinge-cliente-da-cloudflare/
. Acesso em: 15 jul. 2025.

UFRJ. GitHub passou pelo maior ataque DDoS já registrado. Segurança da Informação – TIC/UFRJ, 1 mar. 2018. Disponível em: https://seguranca.tic.ufrj.br/alertas/github-passou-pelo-maior-ataque-ddos-ja-registrado/
. Acesso em: 15 jul. 2025.

VEJA. DDoS: o ataque hacker que derruba redes oficiais e já mirou Bolsa Família. Veja, 9 ago. 2023. Disponível em: https://veja.abril.com.br/coluna/maquiavel/ddos-o-ataque-hacker-que-derruba-redes-oficiais-e-ja-mirou-bolsa-familia/
. Acesso em: 15 jul. 2025.

SERVIÇO PÚBLICO FEDERAL
INSTITUTO FEDERAL DE EDUCAÇÃO, CIÊNCIA E TECNOLOGIA DE PERNAMBUCO
BIBLIOTECA DEPOSITÁRIA – CAMPUS PALMARES

TERMO DE AUTORIZAÇÃO PARA DISPONIBILIDADE DE LIVRO / CAPÍTULO DE LIVRO / ARTIGO OU REA NO REPOSITÓRIO INSTITUCIONAL IFPE

Na qualidade de titular dos direitos de autor da publicação, autorizo ao Repositório Institucional do Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco (RIIFPE) a disponibilizar, através do site https://repositorio.ifpe.edu.br/xmlui/, sem ressarcimento dos direitos autorais, de acordo com a Lei n. 9.610/98, o texto integral da obra abaixo citada, a título de divulgação e de preservação digital da produção científica brasileira, a partir desta data.

Identificação:

Autor*	Everton Juan de Souza
E:mail	ejs34@discente.ifpe.edu.br
Orcid	0009-0004-9428-4792
Link Lattes	https://lattes.cnpq.br/4955548837633620
Título	APRENDIZADO DE MÁQUINA APLICADO À DETECÇÃO DE ATAQUES DDoS COM ABORDAGENS DE INTELIGÊNCIA ARTIFICIAL EXPLICÁVEL (XAI).
Data de defesa	30/07/2025
ISBN	
DOI	
ODS Agenda 2030 (quando cabível)	

*Preenchimento individual, em caso de mais de um autor.

LICENÇA DE DIREITO AUTORAL

Na qualidade de titular dos direitos de autor do conteúdo supracitado, autorizo o Instituto Federal de Educação, Ciência e Tecnologia de Pernambuco a disponibilizar a obra, gratuitamente, de acordo com a licença pública *Creative Commons*, Licença 4.0 *Unported* por mim declarada sob as seguintes condições:

Permitir uso comercial da obra? (x) Sim ( ) Não
Permitir modificações em sua obra?
(x) Sim
( ) Sim, contanto que outros compartilhem pela mesma licença
( ) Não

A obra continua protegida por direito autoral e/ou por outras leis aplicáveis, respeitando inclusive o contrato celebrado entre a editora ou periódico que veicula a mesma. Qualquer uso da obra que não o autorizado sob esta licença ou pela legislação autoral é proibido.

_____ 13/09/2025

Assinatura do Autor Data de autorização.