UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Pedro Ramires Esteves

Aplicação de Inteligência Artificial Explicável no Contexto de Detecção de Intrusão em Dispositivos IoT

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

João Pedro Ramires Esteves

Aplicação de Inteligência Artificial Explicável no Contexto de Detecção de Intrusão em Dispositivos IoT

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

João Pedro Ramires Esteves

Aplicação de Inteligência Artificial Explicável no Contexto de Detecção de Intrusão em Dispositivos IoT

Trabalho de conclusão de curso apresentado à Faculdade de Computação da Universidade Federal de Uberlândia, como parte dos requisitos exigidos para a obtenção título de Bacharel em Ciência da Computação.

Trabalho aprovado. Uberlândia, Brasil, 13 de maio de 2025:

Rodrigo Sanches Miani
Orientador

Diego Nunes Molinos

Fernanda Maria da Cunha Santos

Uberlândia, Brasil
2025

"Só a loucura salva os inquietos de coração"

"Pressa" - menores atos

Resumo

A crescente adoção de dispositivos inteligentes em ambientes como redes domésticas levanta a necessidade de considerações de segurança em seu entorno, assim tornando importantes sistemas que possam ajudar a assegurar a segurança cibernética dos mesmos. Circunscrito a esse desafio, este trabalho investiga a aplicação de técnicas de Inteligência Artificial Explicável (XAI) para promover maior transparência de sistemas de detecção de intrusão (IDSs) em dispositivos Internet das Coisas (IoT). Utilizando do conjunto de dados CICIoT2023, foram feitas análises exploratórias e pré-processamento dos dados, seguida da seleção e treinamento de oito modelos de aprendizado de máquina, dentre os quais *XGBoost* se destacou em termos de performance. Para diferenciação entre tráfego normal e de ataque, alcançou 86% de acurácia, e para separar não somente anomalias, mas também seu tipo, obteve 77%, com ressalvas em categorias minoritárias. Empregou-se a técnica SHAP de explicações para análise de contribuições globais e locais das variáveis nas decisões desses modelos, que revelaram dependências ambíguas e potenciais vieses - o protocolo `https` apresentou influência contraditória, ora elevando a probabilidade de tráfego benigno, ora contribuindo para falsos positivos de ataque; a métrica `min` oscilou entre reforçar decisões de ataque e benigno dependendo do caso. Caminhos futuros apontados direcionam-se no sentido de engenharia de atributos e seleção de variáveis.

Palavras-chave: Internet das Coisas. Inteligência Artificial. Sistemas de Detecção de Intrusão. Aprendizado de Máquina. Inteligência Artificial Explicável. SHAP.

Lista de ilustrações

Lista de tabelas

Lista de abreviaturas e siglas

AM	Aprendizado de Máquina
AP	Aprendizagem Profunda
CIC	*Canadian Institute for Cybersecurity*
CNN	*Convolutional Neural Network*
CSV	*Comma-Separated Values*
DARPA	*Defense Advanced Research Projects Agency*
DDoS	*Distributed Denial of Service*
DoS	*Denial of Service*
HIDS	*Host-based Intrusion Detection System*
HTTP	*HyperText Transfer Protocol*
HTTPS	*HyperText Transfer Protocol Secure*
IANA	*Internet Assigned Numbers Authority*
IDS	*Intrusion Detection System*
IoT	*Internet of Things*
ML	*Machine Learning*
MLP	*MultiLayer Perceptron*
NIDS	*Network-based Intrusion Detection System*
PCAP	*Packet CAPture*
RBF	*Radial Basis Function*
SHAP	*SHapley Additive exPlanations*
SGD	*Stochastic Gradient Descent*
SOMs	*Self Organizing Maps*
UNB	*University of New Brunswick*
XAI	*eXplainable Artificial Intelligence*
X-IDS	*eXplainable Intrusion Detection System*

Sumário

1 Introdução

O uso de Aprendizado de Máquina no âmbito da Segurança da Informação tem se tornado cada vez mais pervasivo na detecção de atividades maliciosas em redes de computadores (HAQ et al., 2015). Nesse contexto, os sistemas de detecção de intrusão (IDS), geralmente posicionados na borda de rede para monitorar fluxos de pacotes, diferenciam tráfego benigno de malicioso e ajudam na tomada de decisões dos times de segurança (UPPAL; JAVED; ARSHAD, 2014).

Para garantir a eficácia desses sistemas, é necessário treinar algoritmos de aprendizado de máquina em conjuntos de dados contendo registros de tráfego de rede. Entretanto, a qualidade desses modelos está diretamente associada aos dados sob os quais eles são treinados, no sentido de que os padrões estatísticos inferidos por eles podem ser compreendidos como consequência direta da estrutura latente desses dados (GONG et al., 2023). Diante disso, compreender o processo de aprendizado — isto é, como os modelos internalizam as características dos dados — sob a ótica de técnicas de Inteligência Artificial Explicável (XAI) constitui o foco desta investigação.

Os classificadores são algoritmos de Aprendizado de Máquina (AM) que, a partir de dados rotulados, aprendem a distinguir entre diferentes categorias (NAQA; MURPHY, 2015) — aqui, tráfego benigno e malicioso. Para a detecção de intrusão em IoT, cada classificador recebe como entrada as características extraídas dos fluxos de rede e gera uma predição sobre seu rótulo.

As técnicas de XAI, então, lançam luz sobre esse processo: elas permitem, por um lado, avaliar a importância média de cada atributo em todas as predições (visão global) e, por outro, detalhar como cada variável enviesou uma decisão individual de uma instância específica (visão local). Ao explorar a maneira como um algoritmo de AM internaliza os padrões presentes nos dados, é possível fazer uma introspecção trazendo a tona justificações para as decisões tomadas. Desse modo, caminha-se para uma direção de maior confiança nos IDSs por parte de seus operadores e partes interessadas, como é discutido em (NEUPANE et al., 2022).

Trabalhos correlatos confirmam que o uso de técnicas de XAI em sistemas IDS é um tema crescente na literatura. Revisões sistemáticas como (NEUPANE et al., 2022) explicitam os principais avanços em X-IDS (XAI aplicada a sistemas IDS), apontando desafios e a necessidade de métodos que conciliem desempenho e transparência.

Diferentes estudos apresentam aplicações pontuais de XAI em IDS tradicionais: Patil et al. (2022) é um exemplo no qual são treinados modelos no conjunto de dados CICIDS2017 (SHARAFALDIN; LASHKARI; GHORBANI, 2018) e exploradas explicações

locais para entender decisões individuais; Sharma et al. (2024) aplicam múltiplas técnicas de explicações em arquiteturas de AM avançadas para IDS genéricos; Roshan e Zafar (2021) usam XAI para analisar detecção de anomalias em tráfego de rede. Esses trabalhos ilustram o valor de abordagens que usam de explicações, mas limitam-se, em sua maioria, a cenários de rede tradicionais, sem focar especificamente em ambientes IoT.

Observa-se, assim, que poucas investigações aprofundam XAI em IDSs voltados a tráfego IoT. Embora existam propostas que usem o conjunto de dados escolhido - CICIoT2023 (NETO et al., 2023) - em configurações sem explicações (THEREZA; RAMLI, 2023) ou somente com análises globais (LE et al., 2023), falta um tratamento integrado que combine explicações globais e locais em uma única metodologia.

Diante dessa lacuna, o presente trabalho tem como objetivo geral tornar mais transparentes as decisões de classificadores treinados no CICIoT2023, para então aplicar explicabilidade utilizando a técnica SHAP (*SHapley Additive exPlanations*) (ROSHAN; ZAFAR, 2021), a fim de discutir vieses e percepções capturadas pelos modelos a partir dos dados. A escolha do conjunto de dados baseia-se em dois eixos: na sua atualidade, e na existência de trabalhos correlatos com (LE et al., 2023) e sem (THEREZA; RAMLI, 2023) análises de explicações. Logo, os objetivos específicos são:

1. Explorar a natureza e estrutura dos dados do CICIoT2023;
2. Comparar o desempenho de diferentes tipos de classificadores;
3. Aplicar XAI global e localmente, tanto em um cenário binário (tráfego normal versus anômalo), quanto multiclasse (diferenciar categorias de ataques);
4. Avaliar como as variáveis de maior impacto se comportam em diferentes instâncias selecionadas;
5. Extrair *insights* a partir das explicações que visem a obtenção de uma melhor compreensão sobre os dados, com base nas internalizações de aprendizado ocorridas.

Quanto à implementação, o código foi desenvolvido na linguagem *Python*, fazendo-se uso de notebooks *Jupyter* (KLUYVER et al., 2016) para a organização das etapas e visualização dos dados, com uso extensivo das bibliotecas `sklearn` (PEDREGOSA et al., 2011) - que contém implementações dos algoritmos de aprendizagem -, e `pandas` (MCKINNEY, 2010), com funcionalidades de manipulação e processamento dos dados. O código pode ser encontrado no endereço <https://github.com/jpramires/tcc>.

A organização do presente trabalho se dá, então, pela seguinte estrutura: no Capítulo 2, é feita uma revisão bibliográfica com a fundamentação teórica e trabalhos correlatos; no Capítulo 3, são delimitados os passos tomados para a execução do método da

investigação proposta; no Capítulo 4, são apresentados os resultados obtidos; e por fim o Capítulo 5 recapitula os passos feitos, passa pelos *insights* obtidos, e discute brevemente limitações e possibilidades para trabalhos futuros.

2 Revisão Bibliográfica

O presente capítulo visa discutir os principais conceitos e tópicos que serão abordados ao longo do trabalho, a fim de mantê-lo auto-contido. Desde dispositivos embarcados, comumente chamados de Internet das Coisas (IoT, do inglês *Internet of Things*), passando por Sistemas de Detecção de Intrusão (IDS, do inglês *Intrusion Detection Systems*), os quais têm visto grande adoção de métodos de Aprendizado de Máquina (AM) (HAQ et al., 2015).

Para além da definição e exemplos de uso de AM em IDS, também é apresentado **Inteligência Artificial Explicável** (XAI, do inglês *eXplainable Artificial Intelligence*) (Barredo Arrieta et al., 2020), tópico central do trabalho, que pode ser entendida não somente como uma área técnica de estudo, mas um movimento que busca alinhar os objetivos da prática de IA com diretrizes éticas (ADAMSON, 2022), sob a luz de que praticantes têm uma responsabilidade ética única de esclarecer ao público mais amplo as limitações do conhecimento sobre o funcionamento de sistemas de IA de caixa-preta.

2.1 Fundamentação Teórica

2.1.1 Internet das Coisas

Segundo Madakam, Ramaswamy e Tripathi (2015), a melhor definição para da Internet das Coisas (IoT, *Internet of Things*) seria 'Uma rede aberta e abrangente de objetos inteligentes que têm a capacidade de se auto-organizar, compartilhar informações, dados e recursos, reagindo e agindo diante de situações e mudanças no ambiente'. Sendo assim, a Internet das Coisas seria composta, em sua essência, pelos dispositivos embarcados (como geladeiras, câmeras de segurança, roteadores *wi-fi*, lâmpadas inteligentes, aspiradores robóticos, etc.) e pelos protocolos que os regem, isto é, a maneira como esses dispositivos inteligentes se comunicam entre si e nas redes em que são instalados.

Devido a natureza desses equipamentos, de uso cada vez mais comum em redes domésticas, a segurança que eles proveem deixa a desejar, e assim criou-se o termo *Internet of Threats* (MENEGHELLO et al., 2019), evidenciando o cuidado que deveria ser tomado pelos administradores (ou usuários comuns) que pretendem fazer uso desses dispositivos 'inteligentes'. Essa preocupação se justifica pelo fato de que muitos dispositivos IoT comerciais, especialmente os de baixo custo, não oferecem mecanismos de segurança robustos, tornando-se alvos fáceis para uma série de ataques cibernéticos, como vazamento de informações sensíveis, negação de serviço (DoS) e acessos não autorizados à rede.

Diante desse cenário, o artigo discute os riscos de segurança nesse setor, analisando

Figura 1 – Laboratório usado para a geração do *dataset* CICIoT2023, contendo diversos dispositivos IoT.
Fonte: (NETO et al., 2023).

vulnerabilidades em protocolos de comunicação amplamente utilizados e destacando ataques reais documentados na literatura. Além disso, os autores enfatizam a importância de considerar a segurança como um aspecto fundamental no desenvolvimento de sistemas IoT, comparando diferentes tecnologias com base em atributos essenciais, como integridade, anonimato, confidencialidade, autenticação e resiliência.

2.1.2 Sistemas de Detecção de Intrusão

No contexto de sistemas computacionais, a segurança é um fator central para que sejam prevenidos ou remediados ataques à confidencialidade dos dados. Com o advento da Internet e o crescimento exponencial de informações sigilosas sendo trafegadas via as redes de computadores (HAQ et al., 2015), a implantação de ferramentas de análise e detecção de atividade maliciosa ganham forma nos Sistemas de Detecção de Intrusão (IDS). Seu propósito central é analisar tráfego de redes e dispositivos, a fim de detectar potenciais atividades maliciosa, assim ajudando a mitigar possíveis ataques e reportando aos operadores responsáveis, garantindo rastreabilidade e observabilidade para os sistemas sendo monitorados (UPPAL; JAVED; ARSHAD, 2014).

No quesito de implementação, sistemas IDS vem ganhando grande adesão de métodos de Aprendizado de Máquina para garantir uma maior acurácia em suas classificações (HAQ et al., 2015). Uma revisão bibliográfica recente compilou estudos publicados entre os anos de 2018 a 2022, evidenciando o uso extensivo de técnicas supervisionadas de AM em IDSs para ganho de performance e redução de falsos positivos (SILVA; SANTOS; OLIVEIRA, 2023). Em decorrência dessa popularidade, diversos sistemas de código aberto de detecção de intrusão foram desenvolvidos (RØDFOSS, 2011), demonstrando que mesmo um componente de atribuição relativamente simples - detectar anomalias - tem extensa aplicabilidade no contexto da segurança da informação.

Rødfoss (2011), em sua dissertação de mestrado, apresenta uma abordagem prática para o funcionamento básico de um IDS, explorando as três vertentes principais: baseados em assinatura, comparando padrões do tráfego com bases de dados de ataques conhecidos; baseados em anomalia, detectando desvios em relação ao comportamento considerado normal; e baseados em especificação, validando as ações da rede contra um conjunto de regras previamente definidas. Ainda de acordo com o autor, a combinação dessas abordagens pode aumentar significativamente a robustez dos mecanismos de detecção.

Para cumprir sua função, IDSs analisam dados como pacotes de rede (endereços IP, portas, protocolos), registros de *logs* e, em alguns casos, o comportamento de aplicações específicas. Eles podem ser posicionados em diferentes pontos da infraestrutura, como atrás de *firewalls*, em zonas desmilitarizadas (DMZ), em segmentos internos da rede ou em pontos de agregação de tráfego, a depender do nível de visibilidade e proteção desejados. A Figura 2 ilustra alguns posicionamentos típicos de sensores IDS em uma rede corporativa, destacando seus propósitos e áreas de cobertura.

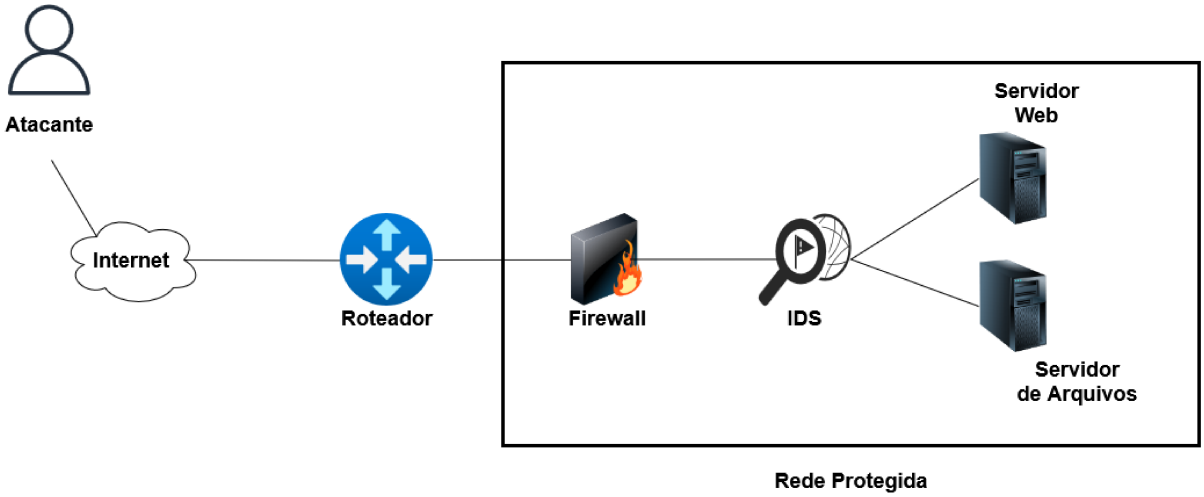


Figura 2 – Exemplo de posicionamento de IDS.
Fonte: Adaptado de (SIADAT; REZVANI; SHIRGAHI, 2016).

2.1.3 Aprendizado de Máquina

O campo de Aprendizado de Máquina (AM, ou ML, de *Machine Learning*) pode ser resumido como uma maneira de programar sistemas computacionais não explicitamente (NAQA; MURPHY, 2015). Em uma era de produção abundante de dados, juntamente com o aperfeiçoamento e consolidação de métodos estatísticos para a inferência de padrões em largas escalas de informação, o AM se vê quase onipresente em todos os campos não só da Computação, mas também em áreas como finanças, entretenimento, biologia, e até mesmo na medicina (SAMUEL, 1959).

Uma possível segmentação da área de AM pode ser feita dividindo-a em categorias dos algoritmos usados (Amazon Web Services, 2023). No Aprendizado Supervisionado, são

providos dados rotulados, que servirão de guia para o modelo (produto final) aprender e inferir padrões. Enquanto isso, no Aprendizado Não-Supervisionado, os dados não tem uma classificação prévia, e a premissa é que sejam criados agrupamentos para os padrões que serão inferidos por meio da semelhança dos dados avaliados.

Dentre as principais tarefas em AM, destacam-se regressão e classificação. A primeira consiste na modelagem da relação entre variáveis de entrada e uma variável contínua de saída, permitindo a predição de valores com base em padrões inferidos a partir dos dados. A segunda, por sua vez, refere-se à atribuição de categorias discretas (duas ou mais) a novas observações. Essas tarefas desempenham um papel central em aplicações práticas e são amplamente descritas em obras canônicas da área, como em (HASTIE; TIBSHIRANI; FRIEDMAN, 2009).

No presente trabalho, o foco será em classificação, abrangendo tanto um cenário binário quanto de múltiplas classes, também grafado como multiclasse. É importante destacar que, em ambos os casos, os modelos de Aprendizado Supervisionado produzem como saída distribuições de probabilidade associadas a cada classe. No binário, a decisão final costuma ser tomada com base em um limiar fixo aplicado sobre a probabilidade atribuída à classe positiva. Já em multiclasse, adota-se normalmente a classe com maior probabilidade como predição final. Essa representação probabilística será especialmente relevante em análises posteriores, onde serão discutidas decisões tomadas em instâncias com probabilidades de alta incerteza, isto é, saídas próximas de 50%.

Em avanços recentes do campo, entra em voga a questão da explicabilidade, visando não apenas a exploração de padrões nos dados, mas também o explícito processo de dedução dos mesmos. Para o atual momento, um bom modelo de Aprendizado de Máquina deve ser capaz de interpretar bem os dados, bem como justificar suas escolhas para seus operadores ou usuários, fazendo uso de métricas que ultrapassam somente as clássicas como precisão ou acurácia, ideia explorada em (MURDOCH et al., 2019).

2.1.3.1 Algoritmos

O emprego de diferentes algoritmos de classificação é central para avaliar como distintas abordagens de AM se comportam no presente cenário. Cada uma das técnicas apresentadas a seguir foi testada neste trabalho, permitindo comparar não apenas o desempenho final, mas também aspectos como eficiência temporal, robustez a ruídos e proporção de falsos negativos. A ordem de apresentação reflete uma progressão de complexidade, desde modelos lineares até métodos baseados em margem.

A *Regressão Logística*, apesar do nome, é um algoritmo de classificação com ideia central de modelar a probabilidade de uma instância pertencer a uma determinada classe, com base em uma combinação linear dos atributos de entrada (COX, 1958). A combinação resultante passa por uma função logística (com formato de S), que comprime o valor para

o intervalo entre 0 e 1, e no caso multiclasse (separação dentre múltiplas categorias), a saída é adaptada por meio da função *softmax*, transformando os valores de saída em probabilidade - isto é, números entre 0 e 1 que, ao serem somados conjuntamente, resultam em exatamente 1.

Uma *Floresta Aleatória* (ou *Random Forest*) é um conjunto (ou *ensemble*) de várias árvores de decisão. Cada árvore é treinada com um subconjunto dos dados e dos atributos, criando diversidade entre elas. Quando da predição, as árvores "votam" entre si e a classe com maioria simples é escolhida (BREIMAN, 2001). Esse mecanismo ajuda a reduzir o risco de sobreajuste típico de uma única árvore, oferecendo previsões mais estáveis e robustas.

O *XGBoost* é um algoritmo que constrói árvores de forma sequencial: a cada nova árvore, o modelo tenta corrigir os erros cometidos pelas anteriores (CHEN; GUESTRIN, 2016). Altamente otimizado, tem foco em velocidade e desempenho, além de incluir mecanismos internos para lidar com valores ausentes, evitar sobreajuste e permitir um controle granular sobre o aprendizado.

Já o *LightGBM* (KE et al., 2017) busca ser ainda mais eficiente em grandes volumes de dados. Uma de suas principais inovações é o crescimento das árvores por folhas, e não por níveis, o que tende a aumentar a acurácia. Além disso, emprega técnicas como histogramas discretos para acelerar o processo de treino e reduzir o uso de memória.

O *MLP* (*MultiLayer Perceptron*) é uma forma simples de rede neural, composta por camadas de neurônios. Cada camada aplica transformações aos dados com pesos ajustáveis e funções de ativação não-lineares. O aprendizado acontece ajustando os pesos para reduzir o erro de cada predição (RUMELHART; HINTON; WILLIAMS, 1986). É uma técnica flexível e capaz de capturar relações não-lineares nos dados.

TabNet é uma rede neural moderna desenvolvida especificamente para dados tabulares, de autoria do Google (ARIK; PFISTER, 2019). Ao contrário de redes comuns, é usado um mecanismo de atenção que aprende a "focar" apenas nas variáveis mais relevantes em cada etapa do processamento. Essa seletividade pode trazer ganhos em interpretabilidade e desempenho, especialmente em tarefas supervisionadas.

A *SVM* (*Support Vector Machine*) procura encontrar o hiperplano que melhor separa as classes no espaço dos atributos. No caso de dados não linearmente separáveis, ela usa um truque de *kernel* para projetar os dados em um espaço onde a separação se torne possível (CORTES; VAPNIK, 1995). O modelo se concentra nas instâncias mais próximas da fronteira de decisão, chamadas vetores de suporte.

Por fim, *SGDClassifier* é uma versão leve e eficiente de classificadores lineares - como Regressão Logística -, porém com treinamento que usa Descida de Gradiente Estocástica (ROBBINS; MONRO, 1951). É adequado para conjuntos de dados grandes

e esparsos, pois processa um número pequeno de amostras por vez. Pode ser rápido e eficiente, mas exige cuidado com a escolha de hiperparâmetros e normalização.

2.1.4 Inteligência Artificial Explicável

Com o advento de métodos de AM que podem entregar resultados com alta taxa de precisão, surge a demanda para que modelos fossem implantados em áreas críticas, devido à sua grande taxa de acerto. Contudo, essa adoção viu críticas voltadas a explicabilidade das decisões tomadas por eles. Em áreas como cardiologia, onde tais sistemas vêm sendo empregados (RUDIN, 2019), é importante compreender *o por quê* de uma decisão ter sido tomada. Nesse aspecto, entra a *eXplainable Artificial Intelligence* - XAI -, concebida pela DARPA (*Defense Advanced Research Projects Agency*) em 2015 para promover maneiras de explicar resultados obtidos por qualquer modelo classificador (GUNNING et al., 2021).

Com explicações, os sistemas podem que empregam componentes baseados em AM são possibilitados de um ganho de confiança tanto dos usuários quanto dos operadores, promovendo assim mais transparência, seja na tomada de decisões - por parte da máquina e dos humanos -; seja no processo dedutivo obtido da interpretação estatística derivada dos dados (Barredo Arrieta et al., 2020). Sob a perspectiva da XAI, as explicações se subdividem em abordagens baseadas em características, perturbação, decomposição e de modelos híbridos (NEUPANE et al., 2022), cada um realizando de diferentes maneiras análises posteriores à classificação a fim de entender os resultados do modelo subjacente sob escrutínio, isto é, chegar a conclusões que possam dar luz ao por quê de determinadas instâncias dos dados terem recebido uma caracterização específica.

Para o presente trabalho, serão adotadas explicações fundamentadas no método SHAP (*SHapley Additive exPlanations*), que utiliza uma abordagem baseada nos valores de *Shapley* para atribuir, de forma consistente, a contribuição de cada característica nas predições, que permitirão tanto análises globais quanto locais (LUNDBERG; LEE, 2017).

2.1.5 SHAP

O SHAP é uma técnica de explicabilidade baseada em características, com fundamentos no conceito de valor Shapley, oriundo da teoria dos jogos cooperativos. Proposto em (SHAPLEY, 1953), o valor de Shapley atribui a cada jogador de um jogo uma parcela justa do ganho total, considerando todas as possíveis coalizões que podem ser formadas entre eles. Essa abordagem foi adaptada para o contexto de modelos preditivos, permitindo a interpretação das contribuições individuais de cada característica na predição de um modelo. Matematicamente, o valor de Shapley para uma característica i é dado por (LUNDBERG; LEE, 2017):

$$\phi_i = \sum_{S \subseteq N \setminus \{i\}} \frac{|S|!(|N| - |S| - 1)!}{|N|!} \left[f_x(S \cup \{i\}) - f_x(S) \right]$$

Onde:

- ϕ_i é a contribuição da característica i;
- N é o conjunto de todos os atributos, com $|N|$ sendo sua cardinalidade (quantidade de elementos);
- S é um subconjunto de N sem o atributo i, com $|S|$ sendo sua cardinalidade;
- $f_x(S)$ é a função aprendida pelo modelo, avaliada no subconjunto S;
- O termo $\frac{|S|!(|N|-|S|-1)!}{|N|!}$ pondera a contribuição da característica i considerando todas as possíveis coalizões.

No contexto de modelos de AM, cada característica (ou atributo) é vista como um 'jogador' que contribui para a predição do modelo. O SHAP calcula a contribuição marginal de cada uma, considerando todas as suas possíveis combinações, chamadas de coalizões. Tal cálculo permite uma decomposição aditiva da predição, ou seja, a soma das contribuições de todas os atributos resulta na predição final do modelo.

Essa abordagem fornece uma explicação consistente (a soma das contribuições de todas as características é sempre igual à diferença entre a predição do modelo e o valor base, independentemente da ordem ou do agrupamento das variáveis) e local para cada predição individual, garantindo que as contribuições sejam proporcionais à influência de cada característica.

Ao considerar todas as possíveis coalizões, o SHAP assegura que a importância atribuída a cada característica seja justa, refletindo sua verdadeira influência no modelo. Esse diferencial é evidenciado por propriedades desejáveis que o método satisfaz (LUNDBERG; LEE, 2017):

- Eficiência: a soma total das contribuições de todas as características corresponde exatamente à predição do modelo, garantindo que nenhum valor seja perdido na explicação;
- Simetria: características que contribuem de maneira idêntica para a predição recebem a mesma importância, independentemente de sua ordem ou posição;
- Jogador Nulo: características que não alteram o resultado em nenhuma coalizão recebem contribuição zero, assegurando que somente variáveis relevantes sejam consideradas.

O SHAP permite a visualização das contribuições das características tanto em nível local quanto global. Para análises locais, gráficos de força (*force plots*) ilustram o impacto individual de cada característica na predição de uma instância específica, enquanto gráficos de dependência (*dependence plots*) ajudam a entender como uma variável influencia as previsões ao longo de seus valores presentes nos dados.

A nível global, gráficos de resumo (*summary plots*) exibem a distribuição dos valores SHAP para todas as instâncias do conjunto de dados, evidenciando quais características exercem maior influência no modelo, enquanto gráficos de barras (*bar plots*) ordenam as médias das importâncias, facilitando a interpretação geral do comportamento do modelo.

Complementarmente, os gráficos locais, como os *force plots*, permitem visualizar a contribuição de cada variável para decisões específicas, oferecendo uma ponte entre a visão agregada e o raciocínio individual do modelo. Essas representações permitem compreender tanto padrões gerais quanto justificativas individuais para as predições.

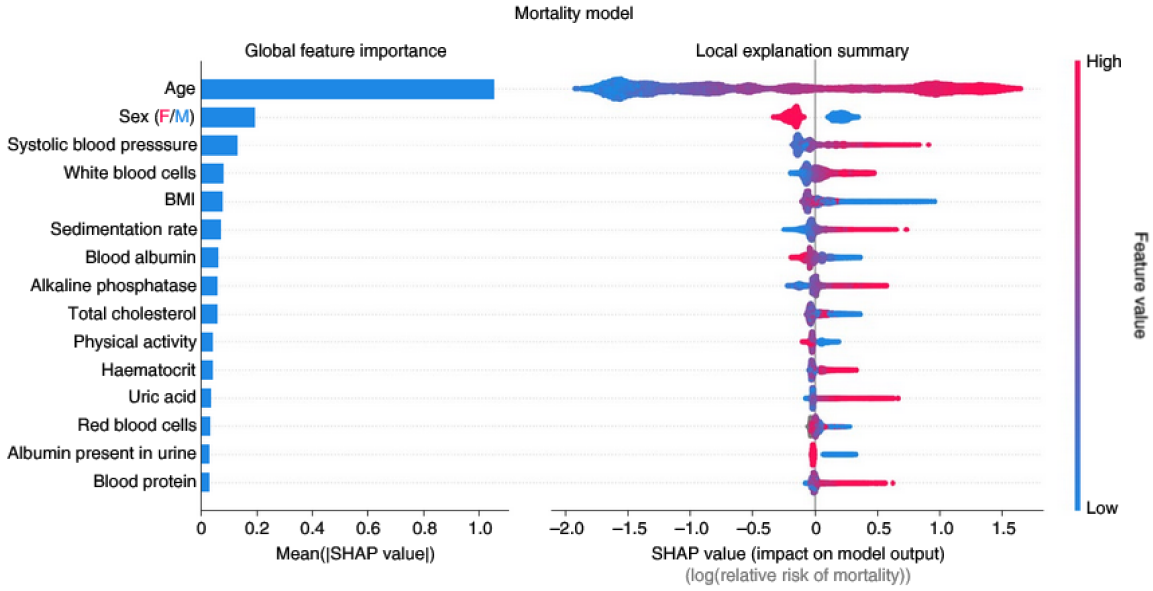


Figura 3 – Exemplo de Gráficos SHAP de Importância Global e Local.
Fonte: Adaptado de (LUNDBERG et al., 2020).

2.2 Trabalhos Relacionados

Ao longo do desenvolvimento deste trabalho, foram elencados artigos correlatos que elucidassem as tecnologias, conceitos e dados que serão utilizados prática e teoricamente, e estes são abordados sucintamente na atual seção. Passando por revisões sistemáticas e proposições de sistemas similares, além de aplicação de XAI, o corpo de trabalhos correlatos serve como contextualização teórica para a metodologia que será desenvolvida no Capítulo 3.

A relação entre XAI e IDS se dá na medida em que esses sistemas podem ser aprimorados ao empregar explicações para suas classificações. Em (NEUPANE et al., 2022), um estudo compreensivo do uso de X-IDS (como é chamada a integração dos dois), os autores reiteram que o emprego de XAI beneficia todos os agentes envolvidos, provendo confiança tanto para os mais externos - como empresários e acionistas -, quanto para os engenheiros diretamente envolvidos na manutenção e operação desses sistemas, como os Analistas de Segurança ou Engenheiros de AM.

Em (NETO et al., 2023), é apresentado e definido o conjunto de dados denominado CICIoT2023. Composto de uma topologia com 105 dispositivos IoT, foram executados 33 ataques de 7 categorias diferentes (*DDoS*, *DoS*, *Recon*, *Web-based*, *Brute Force*, *Spoofing*, e *Mirai*) entre esses dispositivos. Então, todo o tráfego produzido foi registrado com o objetivo de servir como um conjunto de dados realista para o cenário de ataque em redes IoT, facilitando assim a criação e avaliação de modelos de Aprendizado de Máquina por meio da disponibilização dos dados gerados.

(ZARPELÃO et al., 2017) apresenta uma revisão sistemática da literatura sobre Detecção de Intrusão em dispositivos Internet das Coisas. Nele, é apresentada a distinção entre *NIDS* e *HIDS* (*Network-* e *Host-based* IDS, respectivamente) - nos quais a detecção ocorre via monitoramento de tráfego de rede ou também leva em conta informações sobre o Sistema Operacional -, além de discutir abordagens para IDS baseadas em assinatura, anomalia e especificação, cada uma levando em conta diferentes aspectos para realizar a classificação. É destacada a falta de análise aprofundada dos pontos fortes e fracos desses sistemas, juntamente com a baixa diversidade de ataques estudados.

(SHARMA et al., 2024) propõe um Sistema de Detecção de Intrusão Explicável (X-IDS, do inglês *eXplainable Intrusion Detection System*), avaliado nos *datasets* NSL-KDD (TAVALLAEE et al., 2009) e CICIDS2017 (SHARAFALDIN; LASHKARI; GHORBANI, 2018). Para obter explicações locais e globais, são utilizados SOMs (*Self Organizing Maps*), uma técnica de redução de dimensionalidade, além de serem produzidos gráficos relevantes para a interpretação apropriada dos resultados. Esses recursos visuais aprimoram a clareza do processo de tomada de decisão do modelo, tornando-o acessível às partes interessadas.

O trabalho de Barnard, Marchetti e DaSilva (2022) é dividido em dois estágios: primeiro, um classificador *XGBoost* (CHEN; GUESTRIN, 2016) é treinado no conjunto de dados NSL-KDD e são aplicadas explicações SHAP. No segundo estágio, é proposto um IDS do tipo NIDS - baseado em rede -, para avaliar o quanto o modelo consegue distinguir ataques que não estiveram presentes em seu conjunto de treinamento, trabalhando sua generalizabilidade. Ademais, ao fornecer informações sobre a importância dos atributos e processos de tomada de decisão, é facilitada uma abordagem mais proativa e informada à segurança cibernética, levando a um *NIDS* mais resiliente.

Alrefaei e Ilyas (2024) utilizam o conjunto de dados IoT-23 (GARCIA; PARMI-

SANO; ERQUIAGA, 2020) para classificação multiclasse de ataques IoT. Os autores propõem um *pipeline* de detecção baseado em características de fluxo de rede, aplicando modelos de AM tradicionais para categorizar ataques. O estudo priorizou a seleção de atributos já interpretáveis para facilitar a análise pós-detecção, embora sem integrar formalmente técnicas de XAI. Por fim, os autores destacam a viabilidade de modelos leves (que tem tempos baixos de inferência) para IoT, mas apontam a complexidade inerente à heterogeneidade de dispositivos como desafio para a geração de explicações.

Aproximando-se do cenário desta investigação, dois trabalhos recentes se destacam na aplicação do SHAP em sistemas de detecção de intrusão. No primeiro, Mia et al. (2024) propõem uma metodologia baseada em XAI para analisar visualmente erros de classificação em modelos de IDS. Utilizando gráficos SHAP sobrepostos, os autores identificam instâncias de falsos positivos e falsos negativos, críticos nas investigações dos analistas de segurança. A abordagem foi validada com múltiplos conjuntos de dados públicos, demonstrando eficácia na redução de erros de classificação e na melhoria da tomada de decisão.

No segundo, Chen et al. (2024) apresentam uma técnica de seleção de características para IDSs em ambientes IoT, integrando SHAP para realizar a técnica de importância de variáveis. Sua metodologia combina aprendizado profundo com análise de interpretabilidade, permitindo a construção de modelos leves e eficientes sem comprometer a precisão. Os resultados indicam que a seleção de características baseada em SHAP contribui significativamente para a melhoria do desempenho dos modelos.

Em síntese, os trabalhos correlatos abrangem desde revisões sistemáticas de X-IDS e análises de impacto de AM em IDS tradicionais até proposições específicas de metodologias explicáveis em cenários IoT e genéricos. Enquanto estudos como (NEUPANE et al., 2022) e (ZARPELÃO et al., 2017) contextualizam o campo e apontam desafios de diversidade de ataques e falta de transparência, implementações práticas com os conjuntos de dados CICIDS2017 e NSL-KDD demonstram o valor de técnicas de XAI locais e globais, porém sem foco em IoT. Mia et al. (2024) e Chen et al. (2024) aplicam SHAP para diagnóstico de erros e seleção de atributos em IDSs, porém o fazem avaliando modelos genéricos ou metodologias isoladas (aprendizagem profunda ou seleção de variáveis).

Finalmente, o presente trabalho propõe um tratamento integrado para o CICIoT2023, combinando a avaliação de diferentes famílias de algoritmos classificadores, além da análise global e local via SHAP em cenários binário e multiclasse, o que permite não apenas comparar o desempenho de diferentes modelos, mas também investigar como os mesmos atributos impactam decisões em contextos distintos de representação de classes.

3 Desenvolvimento

A metodologia consiste na aplicação de passos sequenciais para a investigação do problema. Delimitada neste capítulo, ela perpassa um caminho linear em direção ao tópico central do trabalho: a aplicação de XAI em um cenário IoT com tráfego malicioso. São descritos aqui os passos que serão tomados a fim de ser feita tal investigação: começando pela seleção dos dados, com sua análise exploratória e pré-processamento; enumeração dos algoritmos classificadores, junto de sua análise de performance e métricas; e por fim a aplicação de SHAP, complementada pelas interpretações que podem ser feitas tomando-as como base.

3.1 Seleção dos dados

A natureza dos dados que o presente trabalho se propõe a analisar está centrada em redes compostas, majoritariamente, por dispositivos IoT, com a presença de tráfego malicioso originário desses próprios dispositivos. Como exemplos correlatos, existem os conjuntos de dados NSL-KDD (TAVALLAEE et al., 2009) e UNSW-NB15 (MOUSTAFA; SLAY, 2015), porém eles não são originalmente concebidos para ambientes IoT, pois focam em redes convencionais e cenários de tráfego genérico.

Para refletir o comportamento dessas redes, é elencado o CICIOT2023 (NETO et al., 2023), disponibilizado pelo CIC/UNB (*Canadian Institute for Cybersecurity*, da *University of New Brunswick*). O tráfego bruto foi inicialmente coletado em arquivos `.pcap`, nos quais diversos dispositivos inteligentes (como sensores e câmeras) interagiam entre si enquanto executavam diversos ataques.

A topologia da rede que originou esses fluxos pode ser vista na Figura 4, na qual observa-se uma malha de 105 dispositivos IoT distribuídos em três sub-redes principais: sensores de ambiente, câmeras de vigilância e atuadores de controle. Cada sub-rede se conecta a um *switch* local, que por sua vez é integrado a um roteador central encarregado de concentrar o tráfego para o servidor de captura. Essa disposição permite simular tanto comunicação máquina-máquina quanto acesso remoto, refletindo cenários reais de uso.

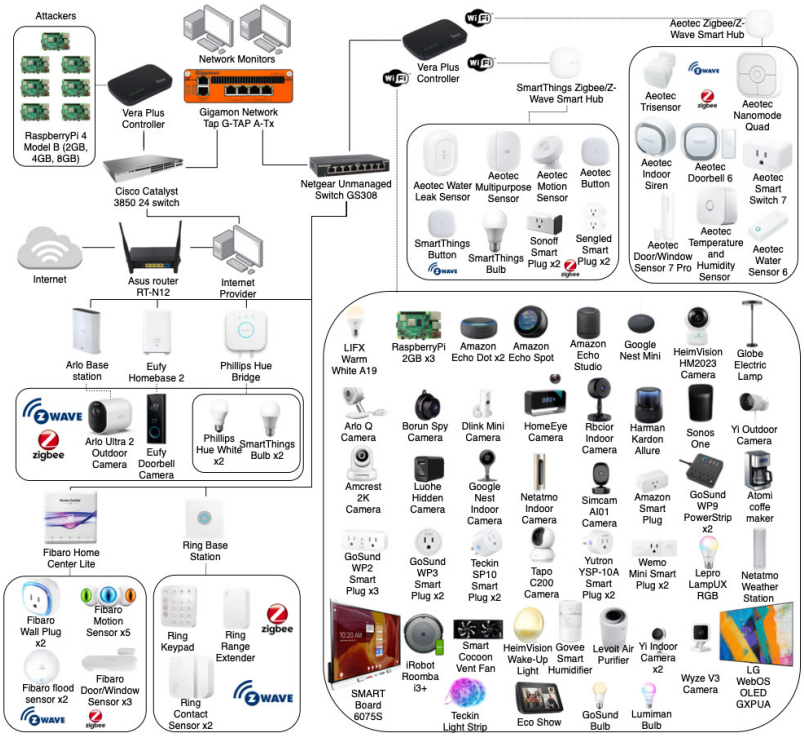


Figura 4 – Topologia da rede usada na geração do CICIoT2023.
Fonte: (NETO et al., 2023).

O entendimento da topologia permite compreender o caráter heterogêneo do tráfego: ataques de negação de serviço podem visar diretamente os *switches*, enquanto varreduras (*recons*) tendem a circular pela malha completa. Ao mapear os pontos de coleta de dados e os caminhos presentes, fundamenta-se a escolha do CICIoT2023 como conjunto de dados, por possuir tanto complexidade topológica quanto diversidade de vetores de ataque.

Utilizaram-se, também, ferramentas customizadas para extrair fluxos de comunicação e gerar tabelas CSV contendo diversas estatísticas de comportamento da rede. Tal transformação padroniza os dados e viabiliza a aplicação de classificadores e técnicas de XAI. Os arquivos individuais podem ser baixados por meio do endereço <http://cicresearch.ca/IOTDataset/CIC_IOT_Dataset2023/>.

Ademais, por haver um número grande de classes (34, contando com o tráfego normal), serão feitas análises de volume e natureza para determinar um subconjunto que possibilite, simultaneamente, derivar interpretações analíticas mais sutis, quando da tarefa de classificação; e obter um corpo de dados relativamente balanceado.

3.2 Pré-processamento dos dados

O pré-processamento de dados é uma etapa essencial em AM, responsável por transformar dados brutos em uma forma mais apropriada para modelagem. Muitas vezes,

os dados coletados diretamente de fontes reais apresentam inconsistências, ruídos, formatos heterogêneos ou lacunas, que podem comprometer o desempenho dos algoritmos. Ao aplicar técnicas de limpeza, transformação e normalização, busca-se mitigar esses problemas, promovendo maior qualidade e representatividade dos dados. Essa fase também é fundamental para garantir que os algoritmos posteriores operem de maneira eficiente, sem vieses artificiais introduzidos pela estrutura subjacente dos dados.

Sendo assim, os dados extraídos do CICIoT2023 serão tratados com o objetivo de garantir maior compatibilidade com os classificadores. Serão aplicados procedimentos de padronização de atributos, tratamento de valores ausentes ou inválidos e correção de falhas de formatação. Além disso, será conduzida uma análise estatística exploratória a fim de compreender preliminarmente a estrutura dos dados e verificar se já há padrões capturáveis visualmente. Serão analisados histogramas de distribuição de ataques, *boxplots* e gráficos de correlação, cujos resultados auxiliarão na interpretação inicial do comportamento das variáveis.

3.3 Seleção dos algoritmos de classificação

Em problemas de classificação, dois cenários principais podem surgir: o binário, no qual há apenas duas categorias possíveis (ataque ou tráfego legítimo); e o multiclasse, no qual cada instância pode pertencer a uma dentre várias categorias — como no presente trabalho, que considera diversos tipos de ataques distintos.

Para a tarefa de classificação multiclasse dos ataques, serão selecionados algoritmos de diferentes famílias, com o objetivo de proporcionar uma comparação entre múltiplas abordagens, como baseadas em árvores, redes neurais e métodos de margem. A seleção contempla desde modelos mais simples até técnicas mais robustas e modernas, e a ordem de aplicação dos algoritmos segue uma progressão de complexidade, permitindo observar como cada categoria se comporta diante do problema.

Serão utilizadas as seguintes implementações: Regressão Logística e *Random Forest*, para uma comparação inicial; *XGBoost* e LightGBM como algoritmos baseados em árvores e *ensembles*; redes neurais, com um *Perceptron* Multicamadas (MLP) e uma Rede Neural Convolucional projetada pelo Google, adaptada para lidar com dados tabulares (*TabNet* (ARIK; PFISTER, 2019)); e, por fim, métodos baseados em margem, com a SVM (*Support Vector Machine*) com kernel RBF (*Radial Basis Function*) e SGDClassifier.

3.4 Treinamento e seleção do melhor modelo

O conjunto de dados será dividido em subconjuntos de treinamento e teste, e será utilizado o método de *holdout*. Este consiste em selecionar aleatoriamente instâncias do

conjunto completo, totalizando uma proporção para que os modelos possam aprender os padrões estatísticos dos dados (sendo este chamado de conjunto de treino), e o resto sendo destinado para a medição da eficácia com que o modelo conseguiu aprendê-los (chamado de conjunto de teste ou validação). A proporção do subconjunto de validação será escolhida como 30%, e um número fixo será passado para o algoritmo amostrador, a fim de garantir reprodutibilidade dos experimentos.

A escolha de reservar 30% dos dados para validação segue uma prática amplamente consolidada na comunidade de AM (GHOLAMY; KREINOVICH; KOSHELEVA, 2018), em que proporções entre 20% e 30% para testes (ou validação) e 70% a 80% para treinamento são recomendadas para equilibrar a necessidade de aprendizado e a avaliação confiável do modelo.

Cada classificador terá seu desempenho testado na detecção das diferentes classes de ataque escolhidas. Assim, será possível fazer uma comparação entre os resultados obtidos por cada um para identificar aquele que melhor generaliza para novos exemplos.

Para a seleção do melhor modelo, serão consideradas métricas de performance que usam das seguintes primitivas: TP (*True Positives*): número de ataques corretamente classificados como ataques; TN (*True Negatives*): número de tráfegos legítimos corretamente classificados como legítimos; FP (*False Positives*): número de tráfegos legítimos incorretamente classificados como ataques; FN (*False Negatives*): número de ataques incorretamente classificados como tráfego legítimo.

Matematicamente, os quatro critérios que serão usados são definidos como:

- **Acurácia**: Proporção de previsões corretas em relação ao total de amostras:

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

- **Precisão**: Proporção de instâncias classificadas como ataque que, de fato, são ataques:

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

- **Revocação (ou Sensibilidade)**: Proporção de ataques corretamente identificados, dentre todos os ataques presentes:

$$\text{Revocação} = \frac{TP}{TP + FN}$$

- **Escore F1**: Média harmônica entre precisão e revocação, útil quando há desequilíbrio entre classes:

$$F1 = 2 \cdot \frac{\text{Precisão} \cdot \text{Revocação}}{\text{Precisão} + \text{Revocação}}$$

Por haverem múltiplas categorias, as métricas para cada escore serão avaliadas segundo uma média simples entre todas as classes, não havendo delegação de peso nem para os fluxos benignos, nem para qualquer ataque em específico.

Além das métricas de desempenho tradicionais, também serão registrados os tempos de treinamento e de teste de cada modelo. Embora esses valores não sejam o foco principal da análise, sua inclusão permite uma visão mais completa sobre os custos computacionais associados a cada abordagem, fornecendo assim uma noção geral de eficiência temporal.

Portanto, será escolhido o modelo que melhor equilibre os quatro escores, com mais peso dado ao mais preciso em termos de Revocação (Sensibilidade), devido à natureza do estudo: para um IDS, é fundamental que sejam minimizados, ao máximo possível, os falsos negativos. Em outras palavras, busca-se capturar a maior proporção de ataques, que, em um cenário real, são esparsos, porém não devem passar despercebidos.

3.5 Aplicação de SHAP no modelo escolhido

Após a seleção, serão considerados dois cenários para a aplicação de técnicas de explicabilidade: o modelo multiclasse original e uma versão ajustada para o caso binário, na qual todas as categorias de ataque serão agrupadas em uma única classe de tráfego malicioso. A distinção entre os dois permitirá avaliar o comportamento do modelo tanto em um cenário mais complexo quanto em um simplificado, sugerindo assim possíveis diferenças em seus critérios de decisão.

Para o modelo binário, serão geradas explicações locais a partir de instâncias selecionadas com base em dois critérios:

1. Casos de baixa confiança na predição, ou seja, decisões próximas ao limiar de decisão, nas quais foram atribuídas probabilidades próximas a 50% para ambas as classes;
2. Casos de alta confiança, mas com classificação incorreta.

Em cada critério, serão analisadas duas instâncias da classe normal e duas da classe de ataque. Além disso, será avaliada a importância global dos atributos, ou seja, sua influência média sobre as decisões do modelo de modo geral.

Para o caso multiclasse, o foco será na análise global. Serão investigados os atributos mais relevantes de forma geral — com a importância calculada como uma média entre todas as classes —, bem como os atributos que mais influenciam em específico as decisões das classes com maior e menor suporte no conjunto de dados. Dessa forma, será possível observar tanto padrões amplos quanto particularidades associadas a essas classes específicas.

A etapa final consiste na análise das explicações obtidas, com o objetivo de interpretar de forma aprofundada o comportamento do modelo diante das instâncias estudadas. Assim, espera-se revelar padrões que ajudem a compreender melhor a natureza do tráfego analisado, além de quais visões os modelos estão internalizando acerca dos dados.

A comparação entre os dois modelos poderá evidenciar diferenças nos critérios decisórios adotados quando o problema é tratado de maneira simplificada e quando envolve múltiplas classes, contribuindo não apenas para a transparência, mas também fornecendo subsídios para uma leitura mais informada sobre o próprios conjunto de dados, auxiliando na identificação de características recorrentes em comunicações maliciosas e reforçando o papel da explicabilidade como uma ponte entre predição e compreensão.

4 Resultados

Este capítulo apresenta os resultados obtidos a partir da execução dos passos descritos no Capítulo 3. São demonstrados a seleção e pré-processamento dos dados, os desempenhos dos modelos testados com as métricas de avaliação, seguidos das análises com a técnica SHAP para as explicações. Também é possível acessar diretamente o código em <https://github.com/jpramires/tcc>.

4.1 Seleção dos dados

Para compreender inicialmente a distribuição das classes, foi realizada uma análise dos rótulos presentes, além da proporção entre tráfego benigno e vetores específicos. Observou-se (Figura 5) que os ataques volumétricos, em especial os de negação de serviço, correspondiam a uma parcela expressiva do conjunto de dados, chegando a compreender quase 90% das amostras rotuladas como maliciosas.

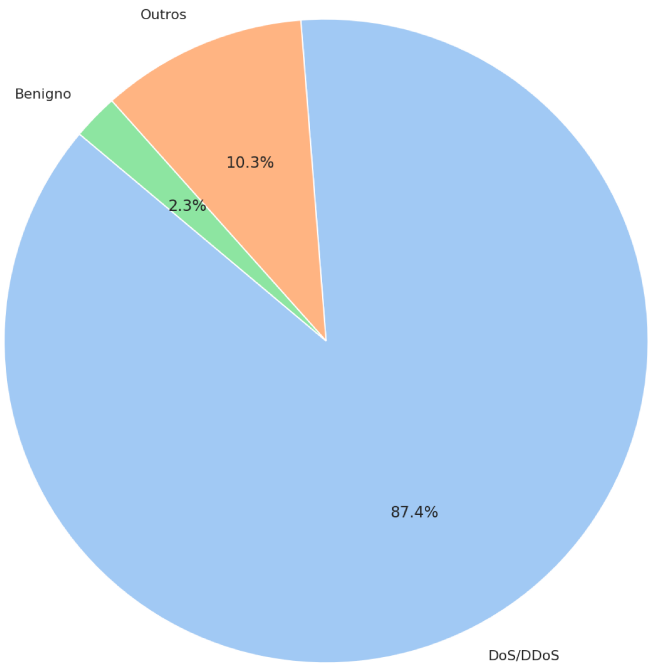


Figura 5 – Proporção de tráfego benigno, ataques volumétricos, e demais vetores.
Fonte: Autor.

Devido ao foco do trabalho ser em ataques que exploram vulnerabilidades de protocolos e aplicações IoT, optou-se por descartar os fluxos associados a estas classes de ataques volumétricos, reduzindo o viés de volume e possibilitando uma análise mais detalhada

dos demais vetores de ameaça. Os detalhes das categorias selecionadas são explicitados na Tabela 1:

#	**Ataque**	**Instâncias**	**%**
1	`VulnerabilityScan`	373,351	30.7 %
2	`MITM-ArpSpoofing`	307,560	25.3 %
3	`DNS_Spoofing`	178,898	14.7 %
4	`Recon-HostDiscovery`	134,378	11.0 %
5	`Recon-OSScan`	98,259	8.00 %
6	`Recon-PortScan`	82,284	6.77 %
7	`DictionaryBruteForce`	13,064	1.07 %
8	`BrowserHijacking`	5,859	0.48 %
9	`CommandInjection`	5,409	0.44 %
10	`SqlInjection`	5,245	0.43 %
11	`XSS`	3,846	0.31 %
12	`Backdoor_Malware`	3,218	0.26 %
13	`Recon-PingSweep`	2,262	0.18 %
14	`Uploading_Attack`	1,252	0.10 %
	Total	1,214,185	100.0 %

Tabela 1 – Distribuição do CICIoT2023.

4.2 Pré-processamento dos dados

Para a manipulação dos dados brutos, primeiramente foi feito um levantamento de quantas entradas haviam de cada classe maior (ataque e normal), que no momento constavam como $1,214,885$ e $1,098,191$, respectivamente. Desse modo, inicia-se com um conjunto balanceado.

Em seguida, transformou-se o atributo "Protocol Type" para uma representação textual, usando do mapa atribuído pela própria IANA (*Internet Assigned Numbers Authority*). A distribuição de protocolos dos fluxos está ilustrada na Tabela 2. Fica clara a predominância dos valores TCP e UDP tanto para o normal quanto ataque, com a adição de que UDP está mais presente para esta segunda classe.

Protocolo	**Benigno**	**Ataque**
TCP	1,004,834.00	942,452
UDP	90,820.00	261,464
HOPOPT	2,302.00	10,155
ICMP	235.00	811
IGMP	Nenhum	3

Tabela 2 – Distribuição de protocolos entre Benigno e Ataque.

Para os valores nulos e inválidos (especificamente `NaN`, *Not a Number*; e `Inf`, Infinito), foram eliminadas as observações que continham tais falhas. Ao todo, subtraiu-se apenas 65 *datapoints* (65 nulos e 0 ruins) benignos e 37 ataques (34 nulos e 3 ruins). Sendo assim, os números finais ficam em $1,214,848$ fluxos anômalos e $1,098,126$ normais.

Quanto à distribuição dos ataques, foi feito um histograma, presente na Figura 6. É possível concluir que as cinco maiores classes têm suporte (quantidade de observações no conjunto) superior a $100,000$, enquanto as menores ficam abaixo de $50,000$, podendo conter apenas alguns milhares ou mesmo centenas de representantes. Fica destacado, então, que há um desbalanço *interno* dentre a classe macro de ataque.

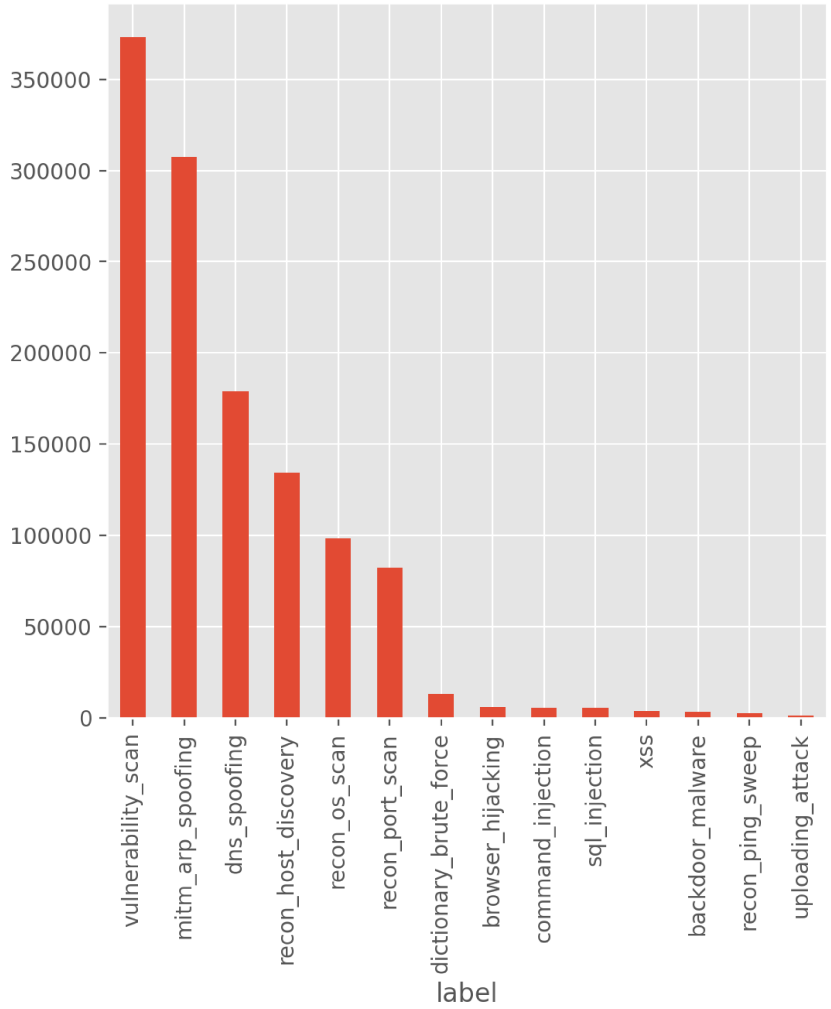
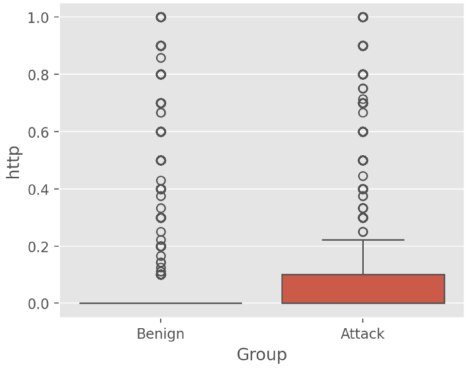


Figura 6 – Distribuição dos Ataques.
Fonte: Autor.

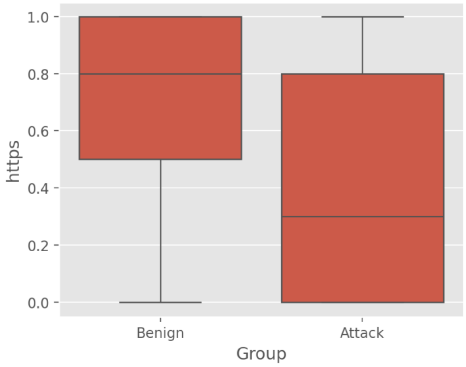
Para a distribuição dos valores dos atributos, é feito o uso de *boxplots*[1]. Havendo um total de 39 atributos, a visualização de todos os gráficos individualmente não agrega

[1] Um *boxplot* (ou diagrama de caixa) é um gráfico que resume a distribuição de um conjunto de dados, exibindo a mediana, os quartis, e possíveis valores atípicos (*outliers*). Útil para visualizar dispersão e simetria dos dados.

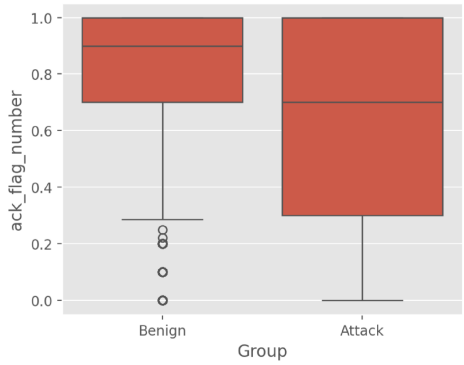
muito à análise, logo são selecionados pares que contenham discrepâncias visualmente significativas, a fim de se discutir diferenças visuais encontradas nesta análise preliminar.



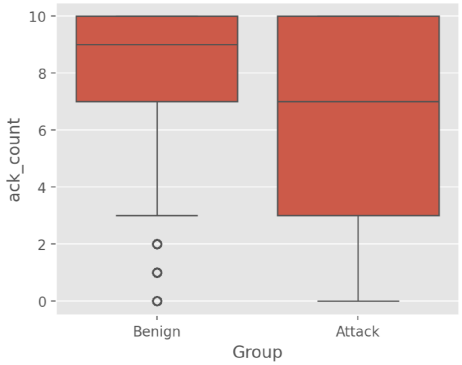
(a) Boxplot do atributo `http`.



(b) Boxplot do atributo `https`.



(c) Boxplot do atributo `ack_flag_number`.



(d) Boxplot do atributo `ack_count`.

Figura 7 – Boxplots dos valores em *features* selecionadas.
Fonte: Autor.

Os gráficos da Figura 7 sugerem que há uma possível separação entre tráfego normal e malicioso, já que algumas *features* apresentam distribuições visualmente distintas — com medianas deslocadas, amplitudes diferentes e outliers em grande volume. Tais padrões *a priori* podem sugerir que há uma estrutura passível de ser destrinchada por um classificador.

Similarmente, nota-se também uma diferenciação nos histogramas das classes macro sobrepostas para os atributos selecionados. Conforme ilustrado na Figura 8, observam-se comportamentos contrastantes entre os fluxos benignos e anômalos, sugerindo que há um potencial discriminativo relevante para a tarefa de classificação.

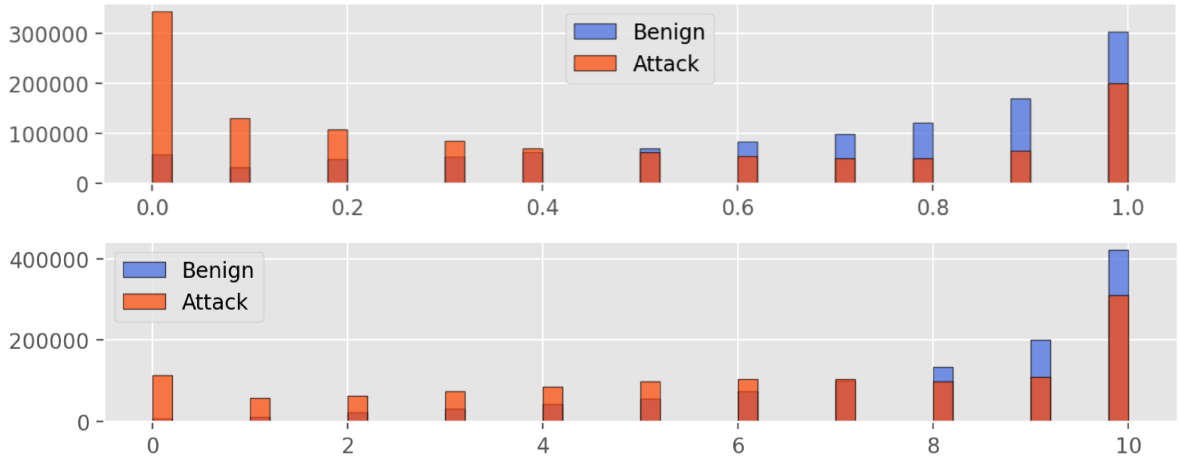


Figura 8 – Histogramas de atributos selecionados
com as classes macro sobrepostas.
Fonte: Autor.

Em seguida, são analisadas as correlações de Pearson entre pares de atributos. Esse coeficiente é definido por $\rho_{X,Y} = \frac{\text{cov}(X,Y)}{\sigma_X \sigma_Y}$, onde $\text{cov}(X,Y) = \frac{\Sigma(x_i - \bar{x})(y_i - \bar{y})}{N-1}$ é a covariância entre as variáveis, σ representa seus desvios padrão e N a quantidade de pontos de dados. A covariância mede a tendência conjunta de duas variáveis se deslocarem em relação às suas médias, isto é, se ambas aumentam ou diminuem simultaneamente (covariância positiva), ou se uma tende a aumentar enquanto a outra diminui (covariância negativa).

Ao ser normalizada pelos desvios padrão, a correlação assume valores entre -1 e 1, indicando o grau e a direção da associação linear. Interpretando os gráficos da Figura 9 e Figura 10 - na qual os coeficientes são retratados em medidas absolutas, desconsiderando a direção -, os resultados sugerem que a aplicação de engenharia de atributos poderia beneficiar análises subsequentes, ao mitigar redundâncias e destacar variáveis mais informativas, já que, visualmente, ao menos 15 e 6 pares apresentam coeficiente igual ou superior a 0.8, evidenciando fortes dependências lineares.

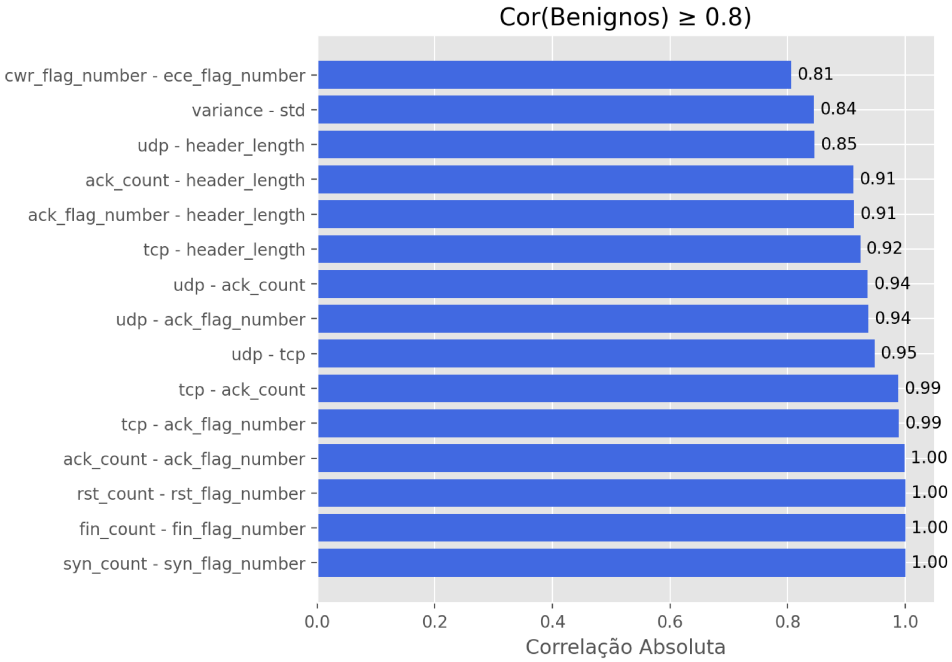


Figura 9 – Pares com $|\rho| \geq 0.8$, para fluxos benignos.
Fonte: Autor.

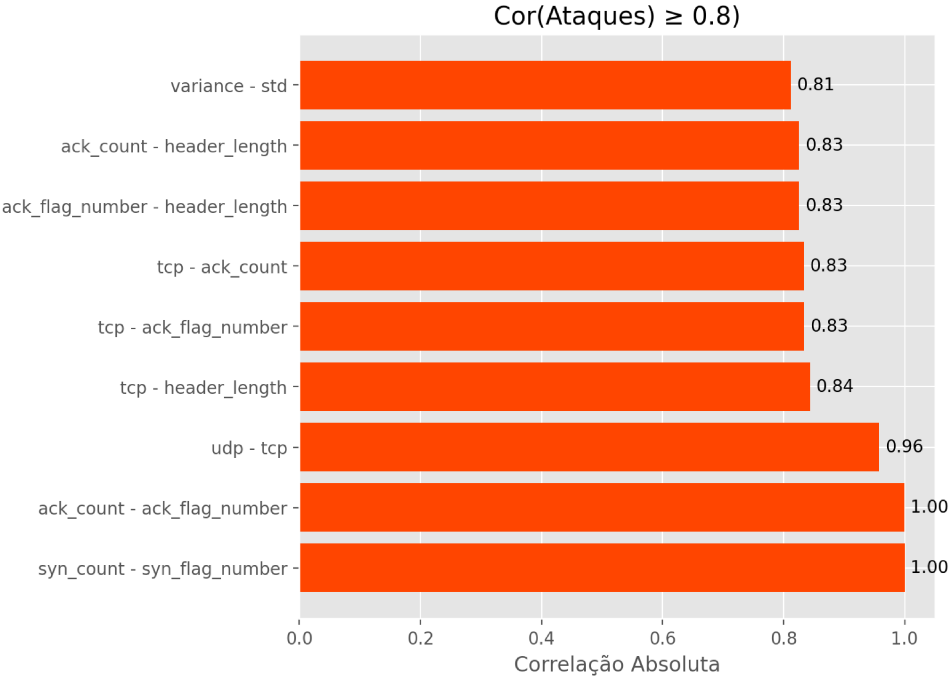


Figura 10 – Pares com $|\rho| \geq 0.8$, para fluxos de ataque.
Fonte: Autor.

Ao fim desta etapa, opta-se pelo agrupamento das categorias de ataque prefixadas com "`recon_`" em uma única classe denominada "`recon`". Tal decisão se justifica pela proximidade conceitual entre esses ataques, todos relacionados a atividades de sondagem e mapeamento da rede.

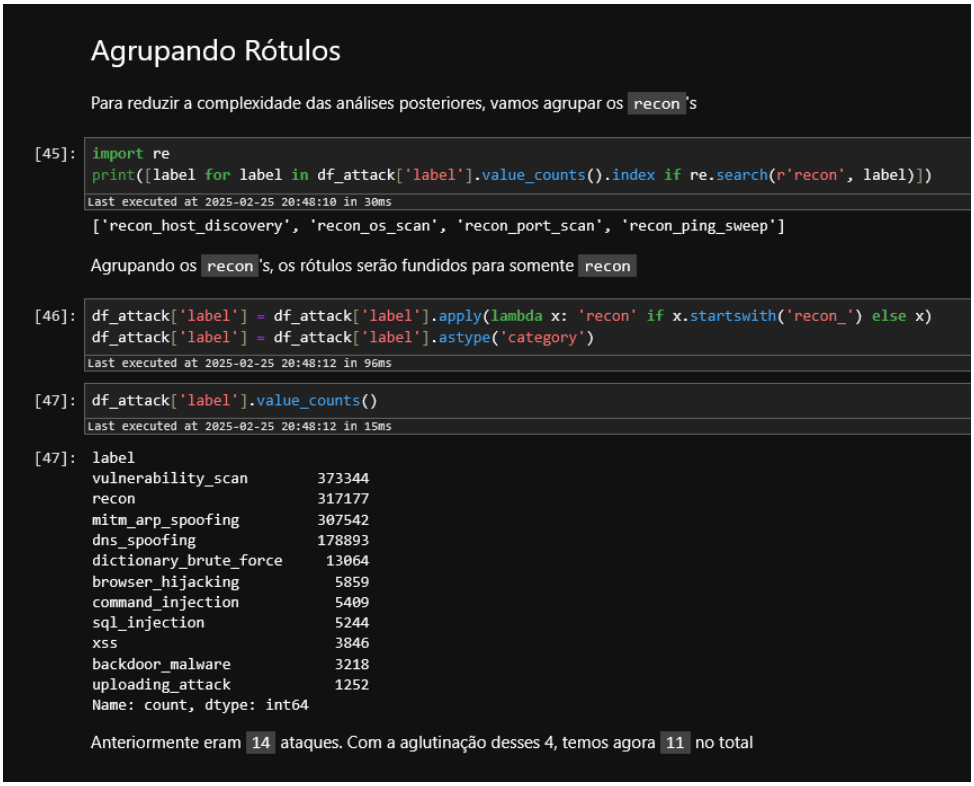


Figura 11 – Agrupamento das categorias de ataque com prefixo `recon_` em uma única classe.
Fonte: Autor.

4.3 Treinamento e seleção do melhor modelo

O conjunto total utilizado para treinamento e avaliação dos classificadores compreende $2,312,974$ observações, cada uma descrita por 40 atributos — 39 variáveis de entrada e 1 variável de saída (denominada `label`), correspondente à classe de tráfego. Já os conjuntos de treino e teste são detalhados conforme a Tabela 3.

Com os dados particionados, todos os modelos foram treinados sobre o mesmo subconjunto de treinamento e avaliados com base no mesmo subconjunto de teste. Para facilitar a análise comparativa, a Figura 12 apresenta, em forma de mapa de calor, um resumo das matrizes de confusão obtidas para cada classificador.

Elas, por sua vez, são uma representação tabular para comparar os valores reais com os valores preditos por cada modelo. Cada linha corresponde às instâncias reais de uma classe, enquanto cada coluna representa as predições feitas, permitindo assim identificar não apenas o desempenho geral, mas também quais classes estão sendo confundidas entre si. Quanto mais claro, maior é o valor retratado.

Classe	**Total**	**Percentual**	**Treino (70%)**	**Teste (30%)**
benign	1,098,126	47.48%	768,688	329,437
vulnerability_scan	373,344	16.14%	261,340	112,093
recon	317,177	13.71%	222,023	95,153
mtm_arp_spoofing	307,542	13.30%	215,279	92,262
dns_spoofing	178,893	7.73%	125,225	53,667
dictionary_brute_force	13,064	0.56%	9,144	3,919
browser_hijacking	5,859	0.25%	4,101	1,757
command_injection	5,409	0.23%	3,786	1,622
sql_injection	5,244	0.23%	3,670	1,573
xss	3,846	0.17%	2,692	1,153
backdoor_malware	3,218	0.14%	2,252	965
uploading_attack	1,252	0.05%	876	375
Total	**2,312,974**	**100%**	**1,619,081**	**693,893**

Tabela 3 – Distribuição das Classes entre Conjuntos de Treino e Teste.

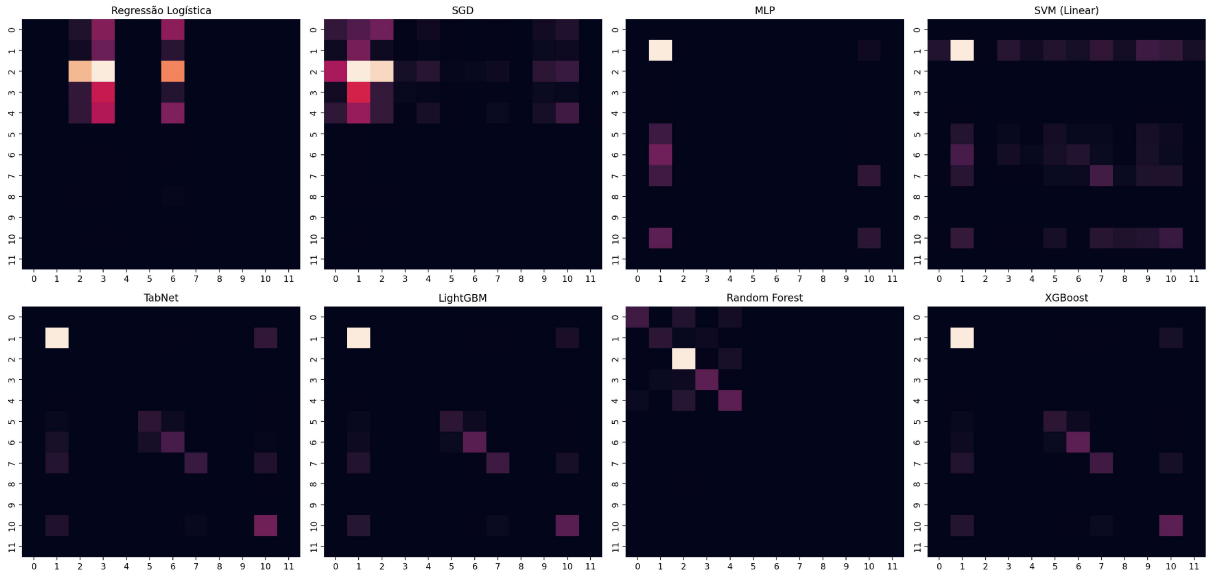


Figura 12 – Matrizes de Confusão dos Classificadores, ordenadas por Revocação Macro: Regressão Logística, SGD, MLP, SVM, *TabNet*, *LightGBM*, *Random Forest* e *XGBoost*.

Fonte: Autor.

Na sequência, a Tabela 4 apresenta os valores das métricas de avaliação para cada modelo testado, permitindo uma comparação direta de desempenho geral, e a Figura 13 traz esses resultados como um gráfico. As entradas estão ordenadas conforme a média de *Recall* Macro, que trata igualmente todas as classes independentemente da sua frequência no conjunto de teste.

Modelo	Acurácia	Precisão	Recall	F1	Treino (s)	Teste (s)
MLP	0.51	0.15	0.11	0.09	225.97	7.21
Regressão Logística	0.25	0.07	0.12	0.06	539.10	0.35
SGD	0.24	0.12	0.14	0.08	244.97	0.36
SVM	0.36	0.17	0.19	0.14	871.56	0.34
TabNet	0.72	0.57	0.32	0.35	4,285.87	7.79
LightGBM	0.76	0.47	0.37	0.40	55.14	10.11
Random Forest	0.77	0.59	0.37	0.41	318.24	23.48
XGBoost	0.77	0.67	0.38	0.43	102.97	3.05

Tabela 4 – Métricas de Desempenho dos Modelos.
Fonte: Autor.

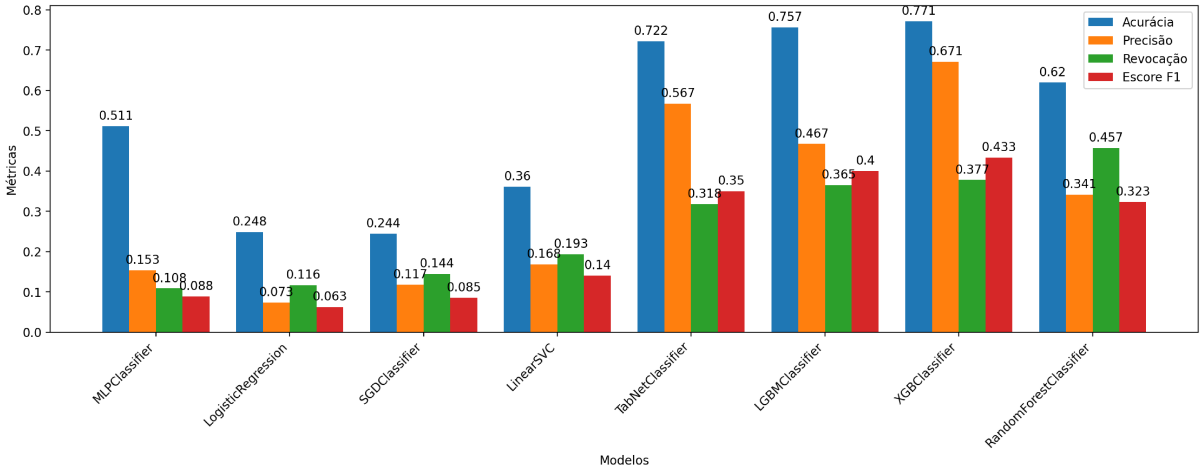


Figura 13 – Métricas Globais por Modelo,
Ordenadas por Revocação Macro.
Fonte: Autor.

De modo geral, os resultados evidenciam diferenças marcantes entre as famílias de algoritmos. Os modelos lineares (Regressão Logística e SGD) apresentaram desempenho limitado, com Revocação e F1 abaixo de 0.15, indicando baixa capacidade de identificar ataques. Por outro lado, os métodos baseados em árvores e *ensembles* (*Random Forest*, *LightGBM* e *XGBoost*) demonstraram ganhos substanciais, superando 0.37 em Revocação e 0.40 em F1. Em particular, o *XGBoost* destacou-se ao alcançar revocação de 0.38 e F1 de 0.43, seguido de perto pelo *Random Forest*, com revocação de 0.37 e F1-score de 0.41.

No que tange à eficiência temporal, observou-se grande variação nos tempos de treinamento: o *TabNet*, por sua complexidade, demandou mais de 4.200 segundos, enquanto o LightGBM e o *XGBoost* completaram o ajuste em cerca de 55 e 103 segundos, respectivamente. Na fase de inferência, todos os ensembles mantiveram tempos de teste abaixo de 25s, com o *XGBoost* executando predições em apenas 3s. Considerando o equilíbrio entre bom desempenho dentre todos os critérios e ênfase em Sensibilidade, o *XGBoost* foi escolhido para a próxima fase, de aplicação e análise de explicações.

Em comparação com os resultados do estudo original do CICIoT2023 (NETO et al., 2023), observam-se diferenças devido à seleção parcial de classes e ao conjunto de algoritmos avaliados. Enquanto os autores do artigo alcançaram acurácia superior a 80% na classificação multiclasse com 34 categorias — chegando a quase 99% em métodos como *Random Forest* —, o experimento presente com *XGBoost* e *Random Forest* resultou em acurácia próxima a 77%.

Adicionalmente, nota-se uma diferença significativa nas taxas de revocação, métrica especialmente relevante neste trabalho por refletir a capacidade do modelo em identificar corretamente ocorrências de cada classe. Enquanto o *Random Forest* no artigo original obteve *recall* de aproximadamente 83%, e o *XGBoost* (não testado pelos autores) é comparável em cenários semelhantes, os melhores resultados aqui alcançados situaram-se na faixa de 38%. Tal dificuldade reforça o desafio imposto pela distribuição específica e limitada de classes selecionadas, que pode dificultar a generalização e cobertura sobre exemplos menos representados.

4.4 Aplicação de SHAP no modelo escolhido

Antes de examinar os valores SHAP, é preciso esclarecer que o *XGBoost* produz, internamente, predições em unidades de *log-odds* (XGBoost Developers, 2023), definidas por:

$$\text{logit}(p) = \log\left(\frac{p}{1-p}\right)$$

Onde p é a probabilidade em relação à classe positiva (ataque). Para converter esses *log-odds* em probabilidade, usa-se:

$$p = \frac{1}{1 + e^{-\text{logit}(p)}}$$

Na prática, um incremento de, por exemplo, 0,7 unidades de *log-odds* corresponde a um aumento na probabilidade de ataque de:

$$\Delta p \approx \frac{1}{1 + e^{-(\text{logit}(p_0)+0{,}7)}} - \frac{1}{1 + e^{-\text{logit}(p_0)}}$$

Onde $\text{logit}(p_0)$ é o valor base (predição média) antes da contribuição. Essa magnitude indica o poder de alteração da crença do modelo: quanto maior o valor absoluto de SHAP em *log-odds*, maior o impacto na probabilidade final. Por exemplo, caso a predição base do modelo para uma instância seja $p_0 = 50\%$, sendo $\text{logit}(p_0) = 0$, e uma variável contribui com 0,7 em unidades de log-odds, a nova predição torna-se $p = \frac{1}{1+e^{-0,7}} \approx 0{,}67$, significando que, isoladamente, tal variável fez a probabilidade do modelo subir de 50% para 67%.

Como as unidades *log-odds* não são imediatamente intuitivas — em especial quando analisadas isoladamente —, a interpretação prática dos valores SHAP tende a ser mais produtiva quando feita de forma comparativa. Em linhas gerais, quanto maiores os valores de SHAP de uma variável, maior é sua influência sobre as decisões do modelo. Assim, a atenção volta-se menos à precisão numérica exata dos valores e mais à identificação de quais atributos exercem maior peso nas predições.

4.4.1 Modelo Binário

Na análise de explicabilidade, ajustou-se uma versão binária do *XGBoost*, na qual todas as categorias de ataque foram agrupadas em uma única classe de tráfego malicioso. A distribuição dos rótulos no conjunto binário ficou conforme apresenta a Tabela 5, e os resultados de avaliação do modelo estão resumidos na Tabela 6. O conjunto utilizado foi o mesmo, então foi mantida a cardinalidade de $2,312,974$ instâncias, sendo $1,619,081$ para treino e $693,893$ para validação.

Rótulo	**Suporte**	**Proporção**
`attack`	1,214,848	52.5%
`benign`	1,098,126	47.5%

Tabela 5 – Distribuição das classes no *dataset* binário.
Fonte: Autor.

Métrica	**Resultado**
Acurácia	0.86
Precisão	0.87
Recall	0.85
Escore F1	0.86
Tempo de Treino	52.77 s
Tempo de Teste	0.73 s

Tabela 6 – Resultados do modelo *XGBoost*.
Fonte: Autor.

Com o modelo binário treinado, parte-se agora para a análise de interpretabilidade, buscando entender quais atributos mais contribuíram para as decisões do modelo.

4.4.1.1 Importância Global

A média dos valores absolutos de SHAP para cada variável, indicada no eixo x do gráfico da Figura 14, corresponde à importância global dessa variável: ela representa o quanto, em média, determinado atributo "move" a predição, em unidades da saída do modelo (LUNDBERG; LEE, 2023), que, no caso do *XGBoost*, são *log-odds*. Valores absolutos maiores indicam que a variável exerce maior influência no modelo, porém sem contar a direção, ou seja, não levando em conta se *aumenta* ou *diminui* a probabilidade de ataque, na crença do classificador.

Também observa-se que a variável `https` tem média absoluta de aproximadamente 0,7, indicando que, em média, ela altera em 0,7 unidades de log-odds a predição do modelo.

As variáveis `min` e `max` — que correspondem, respectivamente, ao tamanho mínimo e máximo de pacotes dentro da janela de fluxo agregada — seguem com importâncias médias de cerca de 0,57 e 0,55. Já a dispersão dos tamanhos de pacote (`std`) e o número de pacotes com a flag `SYN` (`syn_flag_number`) apresentam médias de contribuição em torno de 0,4 e 0,38.

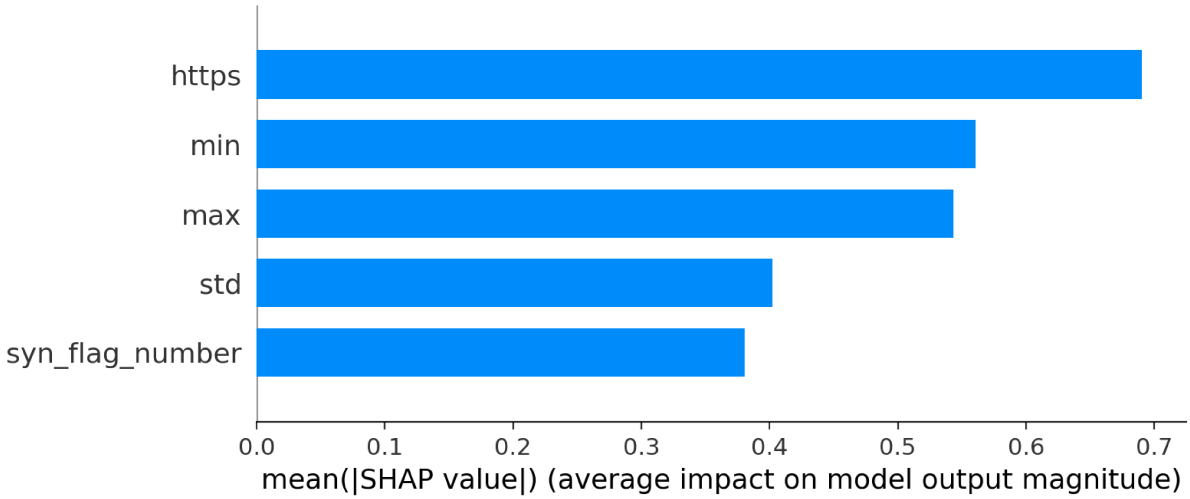


Figura 14 – Importância média das variáveis no modelo *XGBoost* binário.
Fonte: Autor.

Outro modo de visualizar tal resultado é por meio de um *Summary Plot*, representado na Figura 15. Esse tipo de gráfico fornece uma visão detalhada da importância e do efeito de cada variável sobre as predições do modelo, combinando informações de magnitude e direção dos valores SHAP.

No eixo y encontram-se as variáveis de entrada ordenadas decrescentemente por importância, enquanto o eixo x representa os valores SHAP individuais — positivos ou negativos — atribuídos às instâncias. Cada ponto no gráfico corresponde a uma observação do conjunto de teste, e sua cor indica o valor original da variável naquela instância (do azul para valores baixos ao rosa para valores altos).

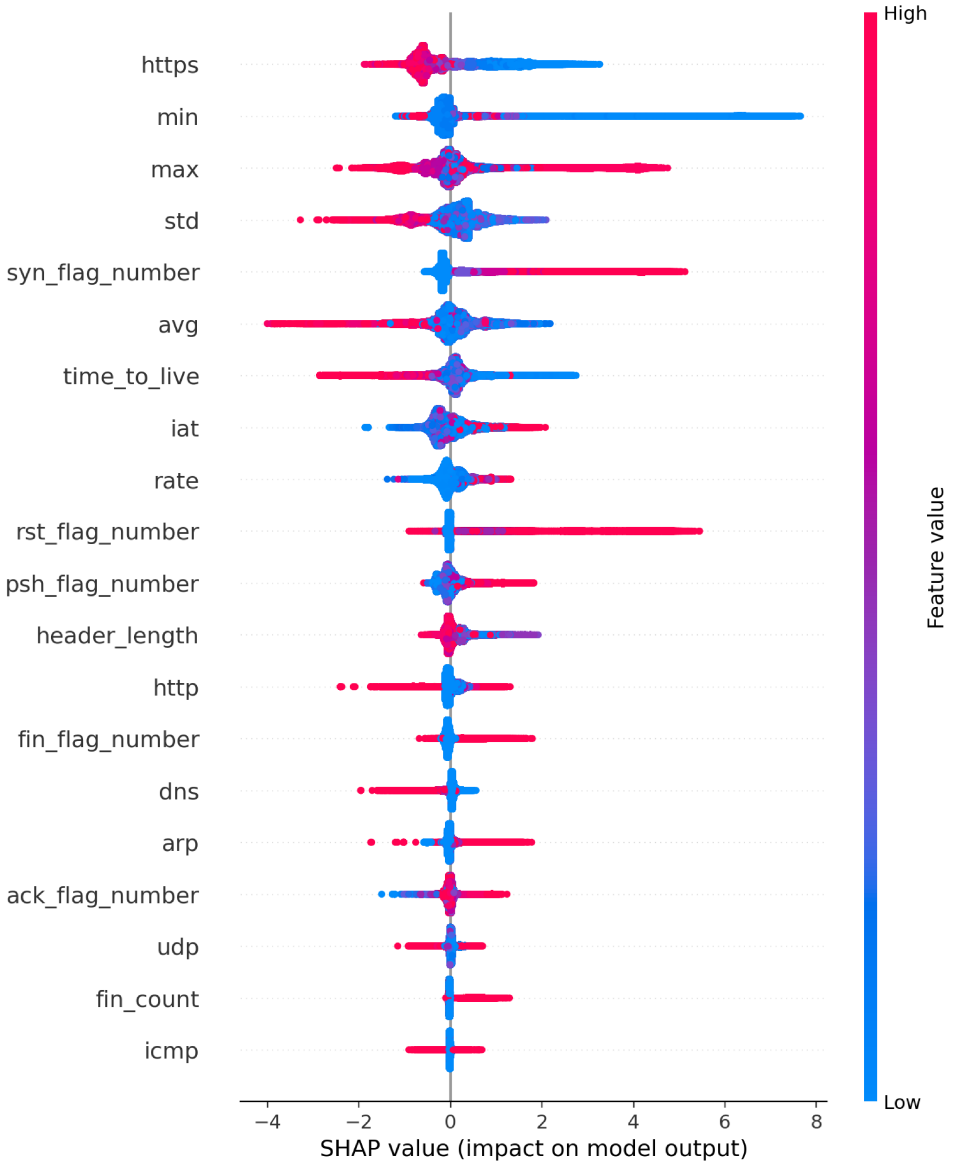


Figura 15 – *Summary Plot* do Modelo Binário.
Fonte: Autor.

A análise dos valores SHAP revela padrões distintos de impacto das variáveis no modelo preditivo. Observa-se uma distribuição heterogênea, com alguns atributos apresentando influência bidirecional. Métricas como `min`, `max` e `rst_flag_number` mostram dispersões que se estendem tanto para o espectro positivo quanto negativo.

As características associadas aos protocolos de comunicação (`https`, `http`, `udp` e `dns`) exibem padrões variados de contribuição. Algumas mostram predominância de impacto em uma direção específica, enquanto outras apresentam distribuição que se estendem para as duas classes.

Métricas estatísticas como `std`, `avg` e `time_to_live` apresentam distribuições relativamente simétricas em ambos os lados, com densidades variáveis. Variáveis como `iat` e `rate` demonstram uma concentração maior perto de 0. Por fim, características relaciona-

das às *flags* de controle TCP (`fin_flag_number`, `psh_flag_number`, `ack_flag_number`) apresentam padrões pouco diferenciados entre si.

Logo, para a tarefa binária, as cinco variáveis mais relevantes (`https`, `min`, `max`, `std` e `syn_flag_number`) demonstram consistentemente maior magnitude de impacto, obtendo distribuições de valores SHAP mais amplas. Tais resultados sugerem que parâmetros relacionados tanto ao estabelecimento de conexões seguras quanto às estatísticas básicas de tráfego são particularmente discriminativos para a tarefa de classificação em questão.

4.4.1.2 Importância Local

Para explorar as decisões individuais, selecionou-se instâncias segundo dois critérios: 1. proximidade da fronteira de decisão, isto é, casos em que a probabilidade prevista esteve próxima de 50%; e 2. classificações incorretas com alta confiança, nas quais o modelo atribuiu probabilidade elevada à classe errada. Em ambos critérios, foram analisadas instâncias preditas nas classes `benign` e `attack`.

O gráfico *Waterfall*, provido pela biblioteca SHAP, permite visualizar como cada atributo contribui para a decisão de uma instância específica. A predição base, $E[f(X)]$, representa a expectativa do modelo antes de considerar os dados da instância, que tem como resultado a saída. A partir desse ponto, cada variável contribui de forma positiva (empurrando para ataque) ou negativa (empurrando para benigno), até se atingir a predição final, $f(X)$. As barras em vermelho indicam aumentos nos *log-odds*, favorecendo a classe positiva, enquanto as azuis indicam reduções, favorecendo a classe negativa.

A seguir, são ilustrados exemplos desses dois cenários, permitindo examinar em detalhe os fatores que influenciaram predições incertas e decisões errôneas.

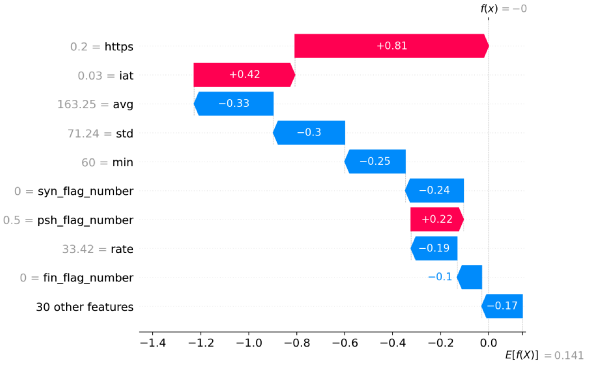


Figura 16 – Gráfico *Waterfall* para instância de baixa confiança predita como `benign`.
Fonte: Autor.

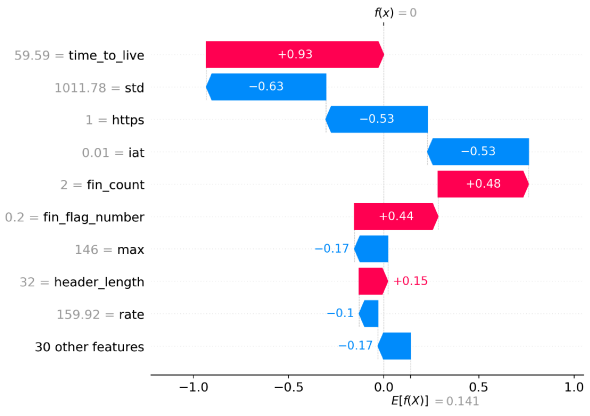


Figura 17 – Gráfico *Waterfall* para instância de baixa confiança predita como `attack`.
Fonte: Autor.

A Figura 16 e a Figura 17 mostram exemplos de instâncias cuja probabilidade

prevista esteve próxima de 50% (pontuação *logit* $f(X) \approx 0$). Em ambos os casos, observa-se um equilíbrio sutil entre forças que puxam a decisão para `benign` e para `attack`, com contribuições do atributo `https` para as duas direções, além da presença de métricas estatísticas - como `avg` e `std` -, e métricas relacionadas ao protocolo TCP.

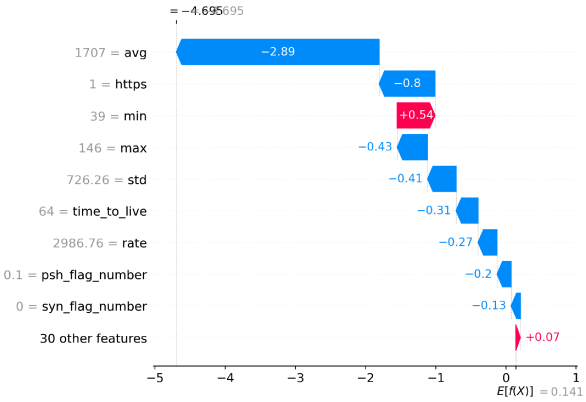


Figura 18 – Gráfico *Waterfall* para instância incorretamente classificada como `benign` com alta confiança.
Fonte: Autor.

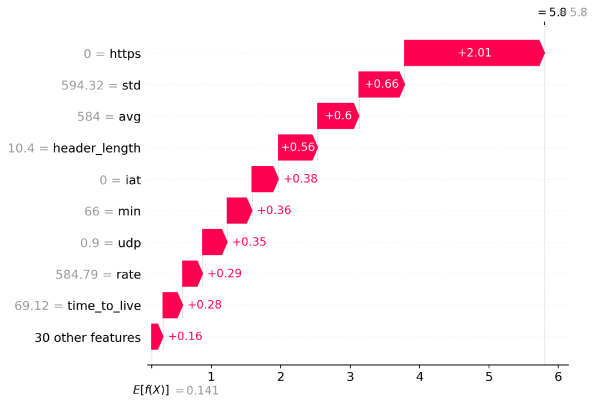


Figura 19 – Gráfico *Waterfall* para instância incorretamente classificada como `attack` com alta confiança.
Fonte: Autor.

Já a Figura 18 e a Figura 19 ilustram casos em que o modelo cometeu erros apesar da alta confiança. Tais exemplos são importantes para investigar possíveis limitações ou padrões incomuns, visto que a combinação das contribuições produziu *log-odds* elevados na direção incorreta. Aqui, encontra-se novamente a presença das medidas estatísticas dos fluxos, além das *flags* de controle TCP, que pendem para a classe 0 (`benign`), apesar do efeito diminuto.

Em síntese, as análises individuais revelam que variáveis como `https`, `avg` e `std` figuram entre as principais fontes de ambiguidade para o modelo nos casos examinados. A variável `https`, em particular, apresentou comportamento contraditório: atuou como forte indicativo da classe `benign` em um caso de fronteira, mas também contribuiu decisivamente para um falso positivo de `attack`.

De modo semelhante, `avg` oscilou entre impactos positivos e negativos expressivos, dependendo da instância analisada. Tal instabilidade sugere que o modelo pode estar atribuindo importância excessiva a essas variáveis de forma descontextualizada, ou ainda que existam interações complexas entre atributos que não estão sendo devidamente modeladas — o que pode explicar decisões altamente confiantes, porém equivocadas.

Vale notar, ainda, que essas variáveis ambíguas coincidem com aquelas que, segundo a análise global, apresentaram os maiores valores médios de importância SHAP. Isso pode indicar uma dependência desproporcional do modelo em relação a atributos que, embora estatisticamente relevantes, carecem de consistência semântica entre as ins-

tâncias. Tal cenário sugere a aplicação, posteriormente, de estratégias de regularização ou de seleção de atributos, práticas especialmente importantes em domínios com forte variabilidade contextual, tal como o tráfego de rede, foco do presente trabalho.

4.4.2 Modelo Multiclasse

Chega-se, por fim, à última etapa da análise interpretativa, dedicada ao modelo multiclasse. Diferentemente do cenário binário, o classificador agora distingue entre múltiplas categorias de tráfego, o que adiciona complexidade às decisões e também às interpretações resultantes.

A distribuição das classes no conjunto de treino pode ser observada na Figura 20. Nota-se uma forte desproporção entre categorias, com poucas classes dominando o volume de instâncias disponíveis. Essa assimetria é um fator importante a se considerar ao avaliar tanto o desempenho do modelo quanto as métricas de explicabilidade.

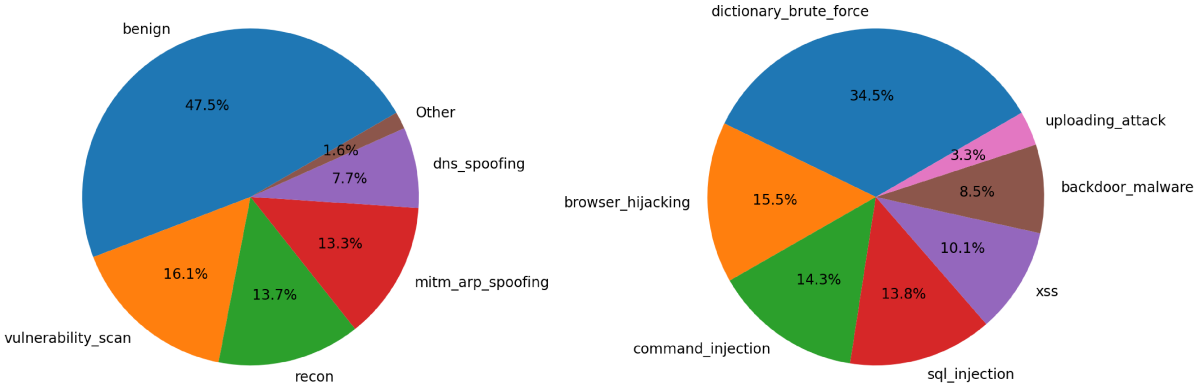


Figura 20 – Distribuição dos Dados de Treino, com Detalhamento da Categoria "*Other*". Fonte: Autor.

4.4.2.1 Importância Global

No contexto multiclasse, os valores SHAP continuam sendo expressos em unidades de *log-odds*, porém são calculados separadamente para cada classe, refletindo o quanto cada variável "empurra" a predição de uma amostra em direção ou ao contrário *de um determinado rótulo*. Ainda que as *log-odds* não sejam diretamente interpretáveis como probabilidades, valores maiores ainda implicam em maior poder de persuasão sobre a saída final do modelo.

A Figura 21 apresenta a importância global das variáveis no modelo multiclasse, obtida a partir da média dos valores absolutos de SHAP, dos quais é tirada novamente a média considerando todas as classes. A visualização oferece uma visão geral sobre quais

atributos exerceram maior influência nas decisões do classificador, independentemente do rótulo para o qual contribuíram.

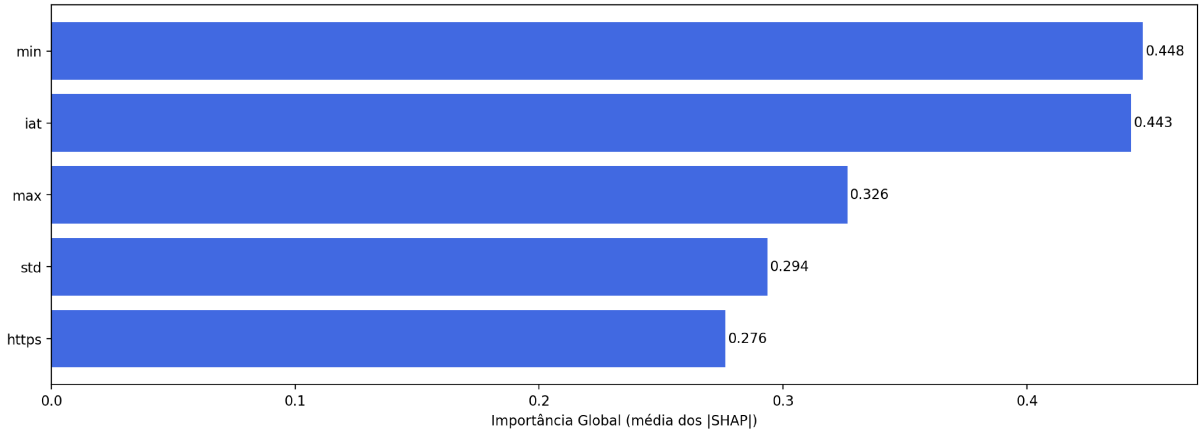


Figura 21 – Importância Global dos Atributos no Cenário Multiclasse.
Fonte: Autor.

Novamente, vê-se a presença de variáveis estatísticas e `https`, porém é notória a introdução de um novo fator: `iat`, a diferença de tempo entre o último pacote registrado. Também percebe-se uma diferenciação no que tange a magnitude dos números, que agora não ultrapassam 0,45, em contrapartida com o cenário binário, no qual se parte de 0,38, chegando até 0,7.

4.4.2.2 Importância Local

Para a análise de importância local no modelo multiclasse, optou-se por examinar as duas classes de ataque com maior e menor número de amostras no conjunto de treino — sendo elas `vulnerability_scan` e `uploading_attack`, conforme ilustrado previamente na Figura 20. Essa escolha visa capturar tanto os padrões dominantes quanto os comportamentos do modelo ao lidar com os casos mais raros, e uma breve descrição de suas performances é dada a seguir, na Tabela 7.

Classe	**Precisão**	**Revocação**	**Escore F1**
`vulnerability_scan`	0.63	0.64	0.64
`uploading_attack`	0.00	0.00	0.00

Tabela 7 – Métricas da Maior e Menor Classe de Ataque.
Fonte: Autor.

Observa-se que o desempenho zerado para a classe `uploading_attack` encontra pistas nas visualizações SHAP, feitas na sequência. Nos *summary plots* locais - como na Figura 23 -, as instâncias desta classe apresentaram valores médios absolutos SHAP

relativamente baixos, indicando que nenhuma variável exerceu papel discriminatório suficientemente forte para essa categoria.

A Figura 22 apresenta o *Summary Plot* da classe com maior representatividade, na qual é possível observar quais atributos mais contribuíram para as predições, bem como sua a direção e dispersão.

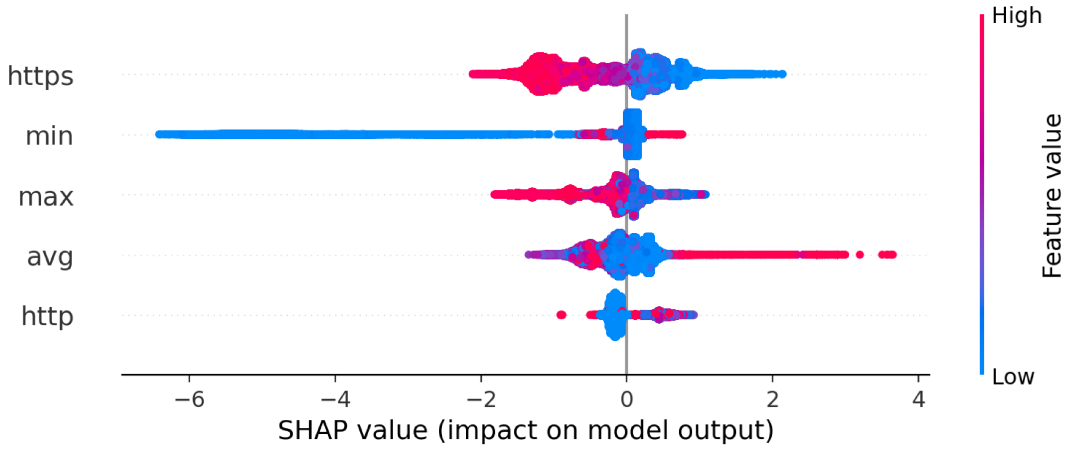


Figura 22 – *Summary Plot* da Classe Maior - *Vulnerability Scan*.
Fonte: Autor.

O atributo `https` mostra um comportamento característico, com valores altos contribuindo negativamente para a predição da classe, enquanto valores baixos empurram a decisão no outro sentido. O padrão indica que fluxos com poucos pacotes HTTPS estão mais associados a essa classe específica. As variáveis estatísticas `min` e `avg` também apresentam dispersões significativas - e em sentidos contrários -, reforçando seu papel como discriminantes importantes para o modelo.

Já a Figura 23 mostra o comportamento da classe menos frequente, revelando um conjunto de atributos parcialmente distinto.

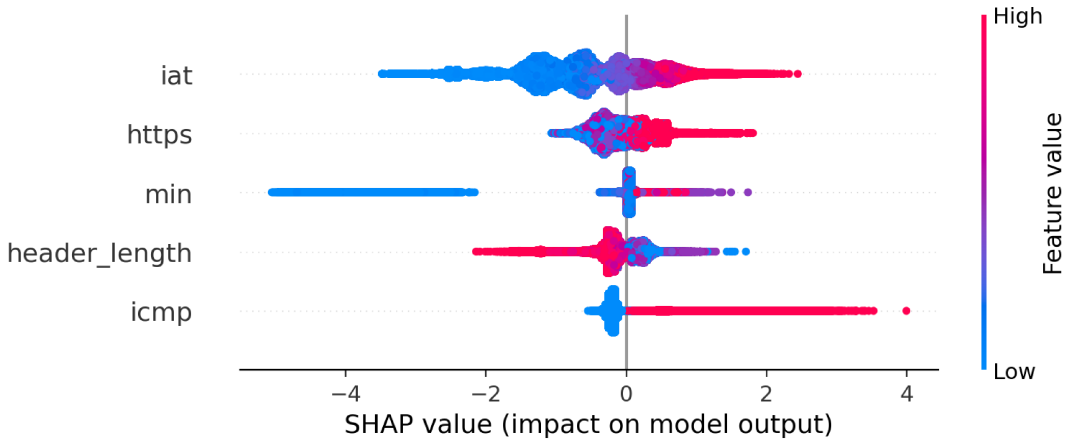


Figura 23 – *Summary Plot* da Classe Menor - *Uploading Attack*.
Fonte: Autor.

Agora, `iat` aparece como a variável de maior impacto, com valores baixos influenciando contra a classe — padrão não observado na classe anterior. `https` permanece relevante, mas com menor intensidade. Outros atributos de destaque incluem `header_length`, que contribui de forma mais polarizada conforme seu valor, e `icmp`, que apresenta uma distribuição assimétrica com fortes pendências para a classe quando toma valores altos.

Ao relacionar essas observações com o gráfico de importância global (Figura 21), nota-se que as variáveis mais recorrentes nas análises locais também figuram entre as de maior impacto médio no modelo multiclasse. A presença de `https`, `min` e `avg` em ambos os contextos reforça seu papel distintivo nas decisões do classificador.

Finalmente, a emergência de atributos como `iat` e `header_length` na classe de poucos representantes sugere que o modelo adapta parcialmente suas estratégias conforme o volume dos dados, mesmo que isso não se traduza em performance, como relata a Tabela 7.

4.5 Considerações e lições aprendidas

Nesta última etapa, sintetiza-se os principais resultados obtidos ao longo deste capítulo, antes de avançar para as conclusões gerais do trabalho.

Resultados Principais

Os experimentos mostraram que, dentre oito classificadores testados, o *XGBoost* apresentou o melhor equilíbrio entre revocação e custo computacional, atingindo aproximadamente 77% de acurácia no cenário multiclasse e 86% no binário. Modelos baseados em árvores (*Random Forest* e *LightGBM*) também se destacaram, enquanto técnicas lineares e SVM ficaram aquém tanto em revocação quanto em escore F1.

A análise SHAP global identificou `https`, `min`, `max`, `std`, `iat` e `header_length` como variáveis de maior influência média, tanto no modelo binário quanto no multiclasse. Já as inspeções locais — com *waterfall* e *force plots* — revelaram que essas mesmas variáveis podem pender decisões de forma contraditória em instâncias de baixa confiança ou em erros confiantes, apontando para zonas de maior ambiguidade no espaço de decisão.

Lições Aprendidas

- **Configuração do dataset:** A exclusão de ataques volumétricos alterou significativamente a complexidade da tarefa, evidenciando que a escolha do subconjunto de classes impacta diretamente nas métricas de avaliação críticas, sobretudo *recall*;

- **Eficácia do *XGBoost*:** O *XGBoost* demonstrou robustez e rapidez na inferência, validando sua adoção como modelo para análise de explicabilidade;
- **Relevância de SHAP integrado:** Combinar análises globais (importância média) e locais (efeitos por instância) proporcionou uma visão abrangente, possibilitando a detecção de padrões gerais e entender com mais exatidão falhas específicas de predição e suas causas;

5 Conclusão

Este trabalho teve como objetivo investigar a aplicação de técnicas de XAI para Detecção de Intrusão em redes de dispositivos IoT, usando o dataset CICIoT2023 como base. Iniciou-se pela análise da natureza e da estrutura dos dados, incluindo a limpeza, o tratamento de valores ausentes, o agrupamento de categorias conceitualmente próximas e o exame das distribuições de classes.

Na sequência, foram selecionados e treinados oito classificadores de diferentes famílias — desde modelos lineares até redes neurais — sobre partições de treino e teste definidas pelo método *holdout* de 30%. As métricas Acurácia, Precisão, Revocação e Escore F1, bem como os tempos de treinamento e de inferência, foram coletados para comparação, e o algoritmo *XGBoost* destacou-se pelo melhor equilíbrio entre capacidade de detecção (*recall*) e precisão, sendo eleito para as etapas de explicabilidade.

A análise global dos valores SHAP revelou, no cenário binário, a predominância de atributos estatísticos e de característicos de rede. No cenário multiclasse, além dessas estatísticas de fluxo e protocolo, emergiram `iat` e *icmp* na classe minoritária. As análises locais, por meio de gráficos *Waterfall* e *Summary Plots*, mostraram como as mesmas variáveis impactantes podem exercer efeitos contraditórios em casos de baixa confiança ou classificações equivocadas.

Como limitação, cabe observar que o estudo foi conduzido em um único experimento sequencial, isto é, após as investigações de explicações não foram feitos novos ajustes ou modificações no *dataset* para se obter modelos mais robustos, práticas que podem ser guiadas partindo-se dos resultados e *insights* obtidos.

Para trabalhos futuros, sugere-se explorar estratégias de engenharia de atributos, como combinações ou transformações da variáveis; comparar diferentes abordagens de XAI, para além de SHAP; e avaliar o desempenho em situações de tráfego real de ataques IoT, de modo a reforçar a robustez e a confiabilidade de sistemas IDS para esses cenários.

Referências

ADAMSON, G. Ethics and the explainable artificial intelligence (xai) movement. **TechRxiv**, 2022. Disponível em: <https://doi.org/10.36227/techrxiv.20439192.v1>. Citado na página 12.

ALREFAEI, A.; ILYAS, M. Using machine learning multiclass classification technique to detect iot attacks in real time. **Sensors**, v. 24, 2024. Citado na página 20.

Amazon Web Services. **What is the difference between supervised and unsupervised learning?** 2023. Disponível em: <https://aws.amazon.com/compare/the-difference-between-machine-learning-supervised-and-unsupervised/>. Citado na página 14.

ARIK, S. O.; PFISTER, T. Tabnet: Attentive interpretable tabular learning. **arXiv preprint arXiv:1908.07442**, 2019. Disponível em: <https://arxiv.org/abs/1908.07442>. Citado 2 vezes nas páginas 16 e 24.

BARNARD, P.; MARCHETTI, N.; DASILVA, L. A. Robust network intrusion detection through explainable artificial intelligence (xai). **IEEE Networking Letters**, v. 4, n. 3, p. 167–171, 2022. Citado na página 20.

Barredo Arrieta, A.; DíAZ-RODRíGUEZ, N.; Del Ser, J.; BENNETOT, A.; TABIK, S.; BARBADO, A.; GARCIA, S.; GIL-LOPEZ, S.; MOLINA, D.; BENJAMINS, R.; CHATILA, R.; HERRERA, F. Explainable artificial intelligence (xai): Concepts, taxonomies, opportunities and challenges toward responsible ai. **Information Fusion**, v. 58, p. 82–115, 2020. ISSN 1566-2535. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1566253519308103>. Citado 2 vezes nas páginas 12 e 17.

BREIMAN, L. Random forests. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 45, n. 1, p. 5–32, out. 2001. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1010933404324>. Citado na página 16.

CHEN, T.; GUESTRIN, C. Xgboost: A scalable tree boosting system. In: **Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining**. New York, NY, USA: Association for Computing Machinery, 2016. p. 785–794. ISBN 9781450342322. Disponível em: <https://doi.org/10.1145/2939672.2939785>. Citado 2 vezes nas páginas 16 e 20.

CHEN, X.; LIU, M.; WANG, Z.; WANG, Y. Explainable deep learning-based feature selection and intrusion detection method on the internet of things. **Sensors**, v. 24, n. 16, 2024. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/24/16/5223>. Citado na página 21.

CORTES, C.; VAPNIK, V. Support-vector networks. **Mach. Learn.**, Kluwer Academic Publishers, USA, v. 20, n. 3, p. 273–297, set. 1995. ISSN 0885-6125. Disponível em: <https://doi.org/10.1023/A:1022627411411>. Citado na página 16.

COX, D. R. The regression analysis of binary sequences. **Journal of the Royal Statistical Society. Series B (Methodological)**, [Royal Statistical Society, Oxford University Press], v. 20, n. 2, p. 215–242, 1958. ISSN 00359246. Disponível em: <http://www.jstor.org/stable/2983890>. Citado na página 15.

GARCIA, S.; PARMISANO, A.; ERQUIAGA, M. J. **IoT-23: A Labeled Dataset with Malicious and Benign IoT Network Traffic**. 2020. Disponível em: <https://doi.org/10.5281/zenodo.4743746>. Citado na página 21.

GHOLAMY, A.; KREINOVICH, V.; KOSHELEVA, O. Why 70/30 or 80/20 relation between training and testing sets: A pedagogical explanation. In: **Computer Sciences Commons**. [s.n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:7467506>. Citado na página 25.

GONG, Y.; LIU, G.; XUE, Y.; LI, R.; MENG, L. A survey on dataset quality in machine learning. **Information and Software Technology**, v. 162, p. 107268, 2023. ISSN 0950-5849. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0950584923001222>. Citado na página 9.

GUNNING, D.; VORM, E.; WANG, J. Y.; TUREK, M. Darpa's explainable ai (xai) program: A retrospective. **Applied AI Letters**, John Wiley & Sons, Inc., v. 2, n. 4, dez. 2021. Disponível em: <https://doi.org/10.1002/ail2.61>. Citado na página 17.

HAQ, N. F.; ONIK, A. R.; HRIDOY, M. A. K.; RAFNI, M.; SHAH, F. M.; FARID, D. M. Application of machine learning approaches in intrusion detection system: A survey. **International Journal of Advanced Research in Artificial Intelligence**, v. 4, 2015. Disponível em: <https://api.semanticscholar.org/CorpusID:15707>. Citado 3 vezes nas páginas 9, 12 e 13.

HASTIE, T.; TIBSHIRANI, R.; FRIEDMAN, J. H. **The Elements of Statistical Learning: Data Mining, Inference, and Prediction**. 2nd. ed. Springer, 2009. ISBN 978-0-387-84858-7. Disponível em: <https://link.springer.com/book/10.1007/978-0-387-84858-7>. Citado na página 15.

KE, G.; MENG, Q.; FINLEY, T.; WANG, T.; CHEN, W.; MA, W.; YE, Q.; LIU, T.-Y. Lightgbm: a highly efficient gradient boosting decision tree. In: **Proceedings of the 31st International Conference on Neural Information Processing Systems**. Red Hook, NY, USA: Curran Associates Inc., 2017. (NIPS'17), p. 3149–3157. ISBN 9781510860964. Citado na página 16.

KLUYVER, T.; RAGAN-KELLEY, B.; PÉREZ, F.; GRANGER, B.; BUSSONNIER, M.; FREDERIC, J.; KELLEY, K.; HAMRICK, J.; GROUT, J.; CORLAY, S.; IVANOV, P.; AVILA, D.; ABDALLA, S.; WILLING, C. Jupyter notebooks – a publishing format for reproducible computational workflows. In: LOIZIDES, F.; SCHMIDT, B. (Ed.). **Positioning and Power in Academic Publishing: Players, Agents and Agendas**. [S.l.], 2016. p. 87 – 90. Citado na página 10.

LE, T.-T.-H.; WARDHANI, R. W.; PUTRANTO, D. S. C.; JO, U.; KIM, H. Toward enhanced attack detection and explanation in intrusion detection system-based iot environment data. **IEEE Access**, v. 11, p. 131661–131676, 2023. Citado na página 10.

LUNDBERG, S.; LEE, S.-I. **shap.Explainer — SHAP latest documentation**. 2023. Disponível em: <https://shap.readthedocs.io/en/latest/generated/shap.Explainer.html>. Citado na página 38.

LUNDBERG, S. M.; ERION, G.; CHEN, H.; DEGRAVE, A.; PRUTKIN, J. M.; NAIR, B.; KATZ, R.; HIMMELFARB, J.; BANSAL, N.; LEE, S.-I. From local explanations to global understanding with explainable ai for trees. **Nature Machine Intelligence**, Nature Publishing Group, v. 2, n. 1, p. 2522–5839, 2020. Citado na página 19.

LUNDBERG, S. M.; LEE, S. A unified approach to interpreting model predictions. **CoRR**, abs/1705.07874, 2017. Disponível em: <http://arxiv.org/abs/1705.07874>. Citado 2 vezes nas páginas 17 e 18.

MADAKAM, S.; RAMASWAMY, R.; TRIPATHI, S. Internet of things (iot): A literature review. **Journal of Computer and Communications**, Scientific Research Publishing, v. 3, n. 5, p. 164–173, 2015. Citado na página 12.

MCKINNEY Wes. Data Structures for Statistical Computing in Python. In: WALT Stéfan van der; MILLMAN Jarrod (Ed.). **Proceedings of the 9th Python in Science Conference**. [S.l.: s.n.], 2010. p. 56 – 61. Citado na página 10.

MENEGHELLO, F.; CALORE, M.; ZUCCHETTO, D.; POLESE, M.; ZANELLA, A. Iot: Internet of threats? a survey of practical security vulnerabilities in real iot devices. **IEEE Internet of Things Journal**, v. 6, n. 5, p. 8182–8201, 2019. Citado na página 12.

MIA, M.; PRITOM, M. M. A.; ISLAM, T.; HASAN, K. **Visually Analyze SHAP Plots to Diagnose Misclassifications in ML-based Intrusion Detection**. 2024. Disponível em: <https://arxiv.org/abs/2411.02670>. Citado na página 21.

MOUSTAFA, N.; SLAY, J. Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In: **2015 Military Communications and Information Systems Conference (MilCIS)**. [S.l.: s.n.], 2015. p. 1–6. Citado na página 22.

MURDOCH, W. J.; SINGH, C.; KUMBIER, K.; ABBASI-ASL, R.; YU, B. Definitions, methods, and applications in interpretable machine learning. **Proceedings of the National Academy of Sciences**, v. 116, n. 44, p. 22071–22080, 2019. Disponível em: <https://www.pnas.org/doi/abs/10.1073/pnas.1900654116>. Citado na página 15.

NAQA, I. E.; MURPHY, M. J. What is machine learning? In: _____. **Machine Learning in Radiation Oncology: Theory and Applications**. Cham: Springer International Publishing, 2015. p. 3–11. ISBN 978-3-319-18305-3. Disponível em: <https://doi.org/10.1007/978-3-319-18305-3_1>. Citado 2 vezes nas páginas 9 e 14.

NETO, E. C. P.; DADKHAH, S.; FERREIRA, R.; ZOHOURIAN, A.; LU, R.; GHORBANI, A. A. Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. **Sensors**, v. 23, n. 13, 2023. ISSN 1424-8220. Disponível em: <https://www.mdpi.com/1424-8220/23/13/5941>. Citado 6 vezes nas páginas 10, 13, 20, 22, 23 e 37.

NEUPANE, S.; ABLES, J.; ANDERSON, W.; MITTAL, S.; RAHIMI, S.; BANICESCU, I.; SEALE, M. **Explainable Intrusion Detection Systems (X-IDS): A Survey of Current Methods, Challenges, and Opportunities**. 2022. Citado 4 vezes nas páginas 9, 17, 20 e 21.

PATIL, S.; VARADARAJAN, V.; MAZHAR, S. M.; SAHIBZADA, A.; AHMED, N.; SINHA, O.; KUMAR, S.; SHAW, K.; KOTECHA, K. Explainable artificial intelligence for intrusion detection system. **Electronics**, MDPI, v. 11, n. 19, p. 3079, 2022. Disponível em: <https://www.mdpi.com/2079-9292/11/19/3079>. Citado na página 9.

PEDREGOSA, F.; VAROQUAUX, G.; GRAMFORT, A.; MICHEL, V.; THIRION, B.; GRISEL, O.; BLONDEL, M.; PRETTENHOFER, P.; WEISS, R.; DUBOURG, V. et al. Scikit-learn: Machine learning in python. **Journal of machine learning research**, v. 12, n. Oct, p. 2825–2830, 2011. Citado na página 10.

ROBBINS, H.; MONRO, S. A Stochastic Approximation Method. **The Annals of Mathematical Statistics**, Institute of Mathematical Statistics, v. 22, n. 3, p. 400 – 407, 1951. Disponível em: <https://doi.org/10.1214/aoms/1177729586>. Citado na página 16.

RØDFOSS, J. T. **Comparison of open source network intrusion detection systems**. Dissertação (Mestrado) — University of Oslo, 2011. Citado 2 vezes nas páginas 13 e 14.

ROSHAN, K.; ZAFAR, A. Utilizing xai technique to improve autoencoder based model for computer network anomaly detection with shapley additive explanation(shap). **International journal of Computer Networks & Communications**, Academy and Industry Research Collaboration Center (AIRCC), v. 13, n. 6, p. 109–128, set. 2021. ISSN 0975-2293. Disponível em: <http://dx.doi.org/10.5121/ijcnc.2021.13607>. Citado na página 10.

RUDIN, C. **Stop Explaining Black Box Machine Learning Models for High Stakes Decisions and Use Interpretable Models Instead**. 2019. Citado na página 17.

RUMELHART, D. E.; HINTON, G. E.; WILLIAMS, R. J. Learning representations by back-propagating errors. **Nature**, v. 323, p. 533–536, 1986. Disponível em: <https://api.semanticscholar.org/CorpusID:205001834>. Citado na página 16.

SAMUEL, A. L. Some studies in machine learning using the game of checkers. **IBM Journal of Research and Development**, v. 3, n. 3, p. 210–229, 1959. Citado na página 14.

SHAPLEY, L. S. **A Value for n-Person Games**. Princeton: Princeton University Press, 1953. 307–318 p. ISBN 9781400881970. Disponível em: <https://doi.org/10.1515/9781400881970-018>. Citado na página 17.

SHARAFALDIN, I.; LASHKARI, A. H.; GHORBANI, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: **International Conference on Information Systems Security and Privacy**. [s.n.], 2018. Disponível em: <https://api.semanticscholar.org/CorpusID:4707749>. Citado 2 vezes nas páginas 9 e 20.

SHARMA, B.; SHARMA, L.; LAL, C.; ROY, S. Explainable artificial intelligence for intrusion detection in iot networks: A deep learning based approach. **Expert Systems with Applications**, v. 238, p. 121751, 2024. ISSN 0957-4174. Disponível em: <https://www.sciencedirect.com/science/article/pii/S0957417423022534>. Citado 2 vezes nas páginas 10 e 20.

SIADAT, S.; REZVANI, M.; SHIRGAHI, H. Proposing a secure method for intrusion detection in amazon ec2 public cloud. **International Journal of Educational Advancement**, v. 7, p. 234–218, 01 2016. Citado na página 14.

SILVA, B. A.; SANTOS, M. P.; OLIVEIRA, T. R. Survey on intrusion detection systems based on machine learning for critical infrastructure protection. **Sensors**, v. 23, n. 5, p. 2415, 2023. Disponível em: <https://www.mdpi.com/1424-8220/23/5/2415>. Citado na página 13.

TAVALLAEE, M.; BAGHERI, E.; LU, W.; GHORBANI, A. A. A detailed analysis of the kdd cup 99 data set. In: **2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications**. [S.l.: s.n.], 2009. p. 1–6. Citado 2 vezes nas páginas 20 e 22.

THEREZA, N.; RAMLI, K. Development of intrusion detection models for iot networks utilizing ciciot2023 dataset. In: **2023 3rd International Conference on Smart Cities, Automation & Intelligent Computing Systems (ICON-SONICS)**. [S.l.: s.n.], 2023. p. 66–72. Citado na página 10.

UPPAL, H. A. M.; JAVED, M.; ARSHAD, M. An overview of intrusion detection system (ids) along with its commonly used techniques and classifications. **International Journal of Computer Science and Telecommunications**, Citeseer, v. 5, n. 2, p. 20–24, 2014. Citado 2 vezes nas páginas 9 e 13.

XGBoost Developers. **Prediction — xgboost documentation**. 2023. Disponível em: <https://xgboost.readthedocs.io/en/stable/prediction.html>. Citado na página 37.

ZARPELÃO, B. B.; MIANI, R. S.; KAWAKANI, C. T.; de Alvarenga, S. C. A survey of intrusion detection in internet of things. **Journal of Network and Computer Applications**, v. 84, p. 25–37, 2017. ISSN 1084-8045. Disponível em: <https://www.sciencedirect.com/science/article/pii/S1084804517300802>. Citado 2 vezes nas páginas 20 e 21.