

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eduardo Ribeiro Gonçalves

**Um Estudo sobre a Generalização de Ataques
de Negação de Serviço em Sistemas de
Detecção de Intrusão**

Uberlândia, Brasil

2025

UNIVERSIDADE FEDERAL DE UBERLÂNDIA

Eduardo Ribeiro Gonçalves

**Um Estudo sobre a Generalização de Ataques de
Negação de Serviço em Sistemas de Detecção de
Intrusão**

Trabalho de conclusão de curso apresentado
à Faculdade de Computação da Universidade
Federal de Uberlândia, como parte dos requi-
sitos exigidos para a obtenção título de Ba-
charel em Ciência da Computação.

Orientador: Rodrigo Sanches Miani

Universidade Federal de Uberlândia – UFU

Faculdade de Computação

Bacharelado em Ciência da Computação

Uberlândia, Brasil

2025

Eduardo Ribeiro Gonçalves

Um Estudo sobre a Generalização de Ataques de Negação de Serviço em Sistemas de Detecção de Intrusão

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, 22 de setembro de 2025:

Rodrigo Sanches Miani
Orientador

Diego Nunes Molinos

Ivan da Silva Sendin

Uberlândia, Brasil
2025

Agradecimentos

Agradeço a minha família por participarem da minha vida, principalmente a minha mãe Taís, responsável por me guiar desde o princípio para o melhor caminho e sempre me apoiar, meu pai José Ricardo e a minha avó Vera, por sempre acreditarem em mim durante essa trajetória.

Agradeço a minha parceira Letícia por sempre estar ao meu lado em todos os momentos e tornar minha vida mais doce.

Aos meus amigos, por tornarem o caminho mais leve. Aos colegas da UFU em que tive o prazer de compartilhar bons momentos e apoio durante o curso, especialmente os da turma 63 da computação.

Por último, agradeço aos professores do curso, por compartilharem os ensinamentos da Ciência da Computação, ao meu orientador Rodrigo Sanches Miani, pela paciência e pelo apoio durante essa jornada e ao Gustavo Di Giovanni Bernardo, por disponibilizar a ferramenta utilizada no desenvolvimento deste trabalho e por sua ajuda.

“..and so he left, with a new tale to tell.” - Risk of Rain 2

Resumo

Com o aumento do tráfego de Internet e, conseqüentemente, dos ataques em redes de computadores, soluções como Sistemas de Detecção de Intrusão (IDS) vêm sendo utilizadas para mitigar esses problemas. Diversos trabalhos na literatura utilizam Aprendizado de Máquina para avaliar se modelos conseguem identificar ataques. Este trabalho analisa modelos treinados com o conjunto CICIDS2017 e com um conjunto experimental em um ambiente de máquinas virtuais, denominado “UFUIDSDOS2025”, contendo tráfego normal e ataques de Negação de Serviço (DoS), investigando suas capacidades de generalização, ou seja, sua habilidade de manter bom desempenho ao classificar dados de cenários distintos daquele em que foi treinado. Foram feitas análises exploratórias, processamento dos dados, e o treinamento de um modelo com o CICIDS2017, utilizando o algoritmo Floresta Aleatória (RF), que obteve altos resultados em métricas como precisão, revocação e medida-F1. Contudo, ao ser avaliado com o conjunto UFUIDSDOS2025, apresentou desempenho muito baixo. Por outro lado, o modelo treinado com o conjunto experimental obteve bons resultados na sua validação, e ao ser validado com amostras do CICIDS2017, conseguiu detectar ataques DoS, embora com alto número de falsos positivos. Para investigar as causas da baixa performance, aplicaram-se técnicas de análise das Árvores de Decisão em ambos os modelos. As análises evidenciaram divergências nos caminhos percorridos pelas árvores ao classificar amostras de cada conjunto, resultando em decisões distintas, associadas a variáveis específicas.

Palavras-chave: Inteligência Artificial, Aprendizado de Máquina, Sistemas de Detecção de Intrusão, Floresta Aleatória, CICIDS2017.

Lista de ilustrações

Figura 1 – Exemplo de funcionamento de um ataque DDoS.	14
Figura 2 – Funcionamento geral do algoritmo Floresta Aleatória.	18
Figura 3 – Visão geral do desenvolvimento do trabalho.	22
Figura 4 – Mapa de calor da matriz de correlação das variáveis do CICIDS2017. . .	36
Figura 5 – Variáveis mais frequentes em pares de alta correlação do CICIDS2017. .	36
Figura 6 – Matriz de confusão do modelo de importância de variáveis.	37
Figura 7 – Importância de variáveis utilizando florestas aleatórias.	38
Figura 8 – Matriz de confusão do modelo do CICIDS2017.	39
Figura 9 – Distribuição de predições do modelo CICIDS2017 no UFUIDSDOS2025. .	41
Figura 10 – Probabilidades de previsão para a classe DoS do modelo do CICIDS2017 no conjunto UFUIDSDOS2025.	41
Figura 11 – Matriz de confusão do modelo UFUIDSDOS2025.	43
Figura 12 – Distribuição de predições do modelo UFUIDSDOS2025 com amostras do CICIDS2017.	44
Figura 13 – Variáveis com mais divergências no modelo do CICIDS2017	46
Figura 14 – <i>Boxplots</i> comparativos das variáveis mais divergentes no modelo CI- CIDS2017.	47
Figura 15 – Variáveis com mais divergências no modelo do UFUIDSDOS2025. . . .	49
Figura 16 – <i>Boxplots</i> comparativos das variáveis de pacotes FIN.	49

Lista de tabelas

Tabela 1 – Relação entre trabalhos correlatos e a proposta do trabalho.	21
Tabela 2 – Variáveis categóricas removidas.	34
Tabela 3 – Variáveis de tempo removidas.	34
Tabela 4 – Distribuição das classes do conjunto de dados CICIDS2017.	35
Tabela 5 – Distribuição das classes após remoção e binarização.	35
Tabela 6 – Resultados da validação cruzada do modelo para importância de variáveis.	37
Tabela 7 – Relatório de classificação do modelo do CICIDS2017.	39
Tabela 8 – Resultados da validação cruzada do modelo do CICIDS2017.	40
Tabela 9 – Relatório de classificação do modelo UFUIDSDOS2025.	42
Tabela 10 – Resultados da validação cruzada do modelo UFUIDSDOS2025.	42
Tabela 11 – Exemplo de divergências entre amostras do CICIDS2017 e conjunto UFUIDSDOS2025 no modelo do CICIDS2017.	45

Lista de abreviaturas e siglas

IDS	<i>Intrusion Detection System</i> (Sistema de Detecção de Intrusão)
RF	<i>Random Forest</i> (Floresta Aleatória)
DoS	<i>Denial of Service</i> (Negação de Serviço)
DDoS	<i>Distributed Denial of Service</i> (Negação de Serviço Distribuída)
HIDS	<i>Host Based Intrusion Detection System</i> (Sistema de Detecção de Intrusão baseado em <i>Host</i>)
NIDS	<i>Network Based Intrusion Detection System</i> (Sistema de Detecção de Intrusão baseado em Rede)
AIDS	<i>Anomaly Based Intrusion Detection Systems</i> (Sistema de Detecção de Intrusão baseado em Anomalias)
SIDS	<i>Signature Based Intrusion Detection Systems</i> (Sistema de Detecção de Instrusão baseado em Assinaturas)
AM	<i>Aprendizado de Máquina</i>
TCP	<i>Transmission Control Protocol</i> (Protocolo de Controle de Transmissão)
UDP	<i>User Datagram Protocol</i> (Protocolo de Datagrama do Usuário)
IP	<i>Internet Protocol</i> (Protocolo de Internet)
FTP	<i>File Transfer Protocol</i> (Protocolo de Transferência de Arquivos)
HTTP	<i>Hypertext Transfer Protocol</i> (Protocolo de Transferência de Hipertexto)
HTTPS	<i>Hypertext Transfer Protocol Secure</i> (Protocolo de Transferência de Hipertexto Seguro)
SSH	<i>Secure Shell</i>
CSV	<i>Comma-Separeted Values</i> (Valores Separados por Vírgulas)
SMOTE	<i>Synthetic Minority Oversampling Technique</i>

Sumário

1	INTRODUÇÃO	11
2	REVISÃO BIBLIOGRÁFICA	13
2.1	Referencial Teórico	13
2.1.1	Ataques cibernéticos	13
2.1.1.1	DoS e DDoS	13
2.1.2	Sistemas de detecção de intrusão (IDS)	14
2.1.3	Aprendizado de máquina	15
2.1.3.1	Aprendizado supervisionado e não supervisionado	16
2.1.3.2	Conjuntos de dados	16
2.1.3.3	Treinamento e testes	16
2.1.4	Random Forest	17
2.2	Trabalhos Relacionados	18
3	DESENVOLVIMENTO	22
3.1	Visão Geral	22
3.2	Seleção dos dados	23
3.3	Pré-processamento do CICIDS2017	24
3.3.1	Remoção de colunas com valores ausentes e linhas duplicadas	24
3.3.2	Variáveis categóricas	25
3.3.3	Remoção de variáveis com risco de viés	25
3.3.4	Normalização	25
3.3.5	Remoção de classes	25
3.3.6	Seleção de variáveis	25
3.3.6.1	Análise de correlação	26
3.3.6.2	Importância de variáveis	26
3.4	Treinamento do modelo do CICIDS2017	27
3.5	Criação do conjunto de dados UFUIDSDOS2025 e teste com o modelo CICIDS2017	29
3.6	Treinamento do modelo UFUIDSDOS2025 e teste com conjunto CICIDS2017	30
3.7	Análise de divergências em árvores de decisão	30
4	RESULTADOS	33
4.1	CICIDS2017 pós pré-processamento	33
4.1.1	Limpeza do conjunto de dados e remoção de classes	33

4.1.2	Análise de correlação	35
4.1.3	Análise de importância de variáveis	37
4.2	Treinamento do modelo do CICIDS2017	38
4.3	Criação do conjunto de dados UFUIDSDOS2025 e teste com o modelo CICIDS2017	40
4.4	Treinamento do modelo do UFUIDSDOS2025 e teste com o CICIDS2017	42
4.5	Análise de divergências em árvores de decisão	44
4.5.1	Divergências no modelo CICIDS2017	44
4.5.2	Divergências no modelo do UFUIDSDOS2025	48
4.6	Discussão e Implicações Práticas	49
5	CONCLUSÃO	51
	REFERÊNCIAS	53

1 Introdução

A rápida expansão da Internet, conectando bilhões de dispositivos e usuários globalmente, transformou profundamente a sociedade, oferecendo acesso instantâneo à informação, serviços digitais e comunicação em tempo real. No entanto, essa mesma conectividade ampliou a superfície de ataque, expondo vulnerabilidades exploradas por agentes maliciosos (LI; LIU, 2021).

Dados corporativos, informações pessoais de usuários e sistemas críticos de infraestrutura hoje dependem fortemente da conectividade oferecida pela Internet, tornando-se potenciais alvos para ciberataques (LI; LIU, 2021). Nesse cenário, a proteção desses dados depende cada vez mais de mecanismos eficazes de detecção e mitigação de ameaças cibernéticas, papel desempenhado, em grande parte, pelos Sistemas de Detecção de Intrusão (IDS).

Os IDS desempenham um papel crucial na segurança de redes, monitorando atividades suspeitas e identificando possíveis ataques ou comportamentos maliciosos. Esses sistemas são classificados em duas categorias principais: os IDS baseados em assinatura (SIDS) e os IDS baseados em anomalia (AIDS). Os SIDS identificam intrusões conhecidas comparando padrões com uma base pré-existente, oferecendo alta precisão, mas falhando diante de ataques inéditos. Já os AIDS detectam desvios em relação ao comportamento esperado, recorrendo a modelos estatísticos, heurísticas ou algoritmos de aprendizado de máquina, sendo mais aptos a identificar ameaças novas, embora suscetíveis a maiores taxas de falsos positivos (KHRAISAT et al., 2019).

Na última década, surgiram diversas propostas para o uso de modelos de aprendizado de máquina para detecção de intrusões, com o intuito de identificar novos ataques (VISHWAKARMA; KESSWANI, 2023). Um dos conjuntos de dados mais utilizados para o estudo e desenvolvimento de modelos de aprendizado de máquina voltados à detecção de intrusões é o CICIDS2017 (SHARAFALDIN; LASHKARI; GHORBANI, 2018). O *dataset* foi projetado para simular um ambiente realista de tráfego de rede, contendo tanto atividades benignas quanto uma ampla variedade de ataques, incluindo DoS, DDoS, *Brute Force*, *Botnet*, *Infiltration* e *Web Attacks*.

A qualidade dos conjuntos de dados é um ponto crítico no desenvolvimento e avaliação de IDS. Pesquisadores apontam que os *datasets* devem representar com precisão o tráfego real ou serem específicos para o ambiente de implantação, o que é desafiado pela diversidade das infraestruturas de tecnologia da informação. Mesmo após décadas, muitos *benchmarks* ainda não refletem adequadamente o problema real da detecção de intrusão, especialmente no contexto de modelos baseados em aprendizado de máquina

(SCHRÖTTER; NIEMANN; SCHNOR, 2024).

Diante desse contexto, este trabalho tem como objetivo geral investigar a capacidade de generalização de um modelo treinado com o conjunto CICIDS2017 frente a dados coletados em um ambiente distinto, composto por máquinas virtuais, configuradas com papéis de vítima e atacante. Nesse ambiente, serão reproduzidos ataques iguais aos presentes no conjunto original e assim, um novo *dataset*, denominado “UFUIDSDOS2025”, será criado. Como objetivos específicos, propõe-se:

1. A comparação do desempenho e capacidade de generalização desse modelo com outro, treinado diretamente em amostras coletadas no ambiente de máquinas virtuais. Cada modelo será avaliado não apenas em seu próprio conjunto de dados, mas também no conjunto utilizado para o treinamento do outro, permitindo uma análise cruzada.
2. A aplicação de técnicas de análise de árvores de decisão do algoritmo Floresta Aleatória em ambos os modelos, a fim de identificar variáveis determinantes nas classificações e evidenciar diferenças estruturais entre os padrões de tráfego presentes nos dois *datasets*.

A implementação do código deste trabalho foi feita utilizando a linguagem *Python* com *notebooks Jupyter*, por contar com uma melhor organização e desenvolvimento iterativo. O código está disponível em <<https://github.com/Eduardorib/Generalizacao-IDS>>.

A organização do trabalho seguirá a seguinte estrutura: no Capítulo 2, é abordado o referencial teórico necessário e trabalhos relacionados; o Capítulo 3 trata do desenvolvimento deste trabalho, demonstrando os passos seguidos para chegar ao objetivo explicitado; no Capítulo 4, são mostrados os resultados dos experimentos; e, por fim, no Capítulo 5, as conclusões e sugestões para trabalhos futuros.

2 Revisão Bibliográfica

Este capítulo apresenta os conceitos fundamentais que embasam o desenvolvimento deste trabalho. Inicialmente, são discutidos os ataques cibernéticos, com ênfase nos ataques de negação de serviço (DoS) e negação de serviço distribuídos (DDoS), foco deste estudo. Em seguida, aborda-se o funcionamento dos Sistemas de Detecção de Intrusão (IDS), destacando seus tipos, características e relevância para a segurança de redes. Por fim, são explorados os princípios do aprendizado de máquina e, de forma mais detalhada, o algoritmo *Random Forest*, escolhido como base para o treinamento dos modelos propostos e para análise de suas respectivas árvores de decisão.

2.1 Referencial Teórico

2.1.1 Ataques cibernéticos

Por conta da rápida expansão da Internet, um volume crescente de informações pessoais, corporativas e organizacionais passou a estar disponível nesse ambiente. Esse cenário torna a rede um alvo cada vez mais atraente para ataques, cujo objetivo pode variar desde o roubo de dados até a execução de outras ações prejudiciais. Há diversas definições para ataques cibernéticos na literatura, mas todos tem em comum o objetivo de comprometer a confiabilidade, integridade e disponibilidade de dados (BENDOVSKI, 2015). Entre as famílias de ataques mais comuns, amplamente utilizadas em conjuntos de dados públicos, destacam-se: ataques de negação de serviço (DoS e DDoS), força bruta e ataques a aplicações web (SHARAFALDIN; LASHKARI; GHORBANI, 2018).

2.1.1.1 DoS e DDoS

Os ataques de negação de serviço (DoS) e negação de serviço distribuído (DDoS) têm como objetivo comprometer a disponibilidade de sistemas, sobrecarregando-os. No caso de ataques DDoS, múltiplas máquinas participam em conjunto para aumentar o impacto. Essas ações maliciosas podem se manifestar por meio da saturação da largura de banda, do esgotamento de conexões ou do consumo excessivo de recursos computacionais, como CPU e memória, afetando diretamente a disponibilidade de serviços para usuários legítimos (DOULIGERIS; MITROKOTSA, 2004).

Estudos indicam que mesmo soluções de segurança tradicionais, como sistemas de detecção de intrusões (IDS), podem falhar em detectar ataques distribuídos, os quais podem gerar tanto efeitos diretos, ao atingir um alvo específico, quanto efeitos indiretos, ao impactar serviços colaterais. Em ambientes em nuvem, por exemplo, tais ataques podem

não apenas comprometer o desempenho de aplicações, mas também redistribuir a carga de forma a propagar o impacto dentro da infraestrutura compartilhada (BONGUET; BELLAICHE, 2017).

Os estratégias dos ataques podem variar. Entre elas, destacam-se: a sobrecarga de recursos, como nos ataques *SYN flood*, que provocam o esgotamento de conexões TCP; a saturação de largura de banda, como em *UDP floods*, que exploram gargalos de rede; a sobrecarga de computação, caracterizada pelo uso massivo de instâncias em plataformas ou pelo envio de *payloads* de grande tamanho para consumir recursos de CPU e memória; e a exploração de vulnerabilidades, por meio do envio de pacotes maliciosos específicos capazes de causar falhas ou lentidão, mesmo sem a necessidade de alta taxa de tráfego (BONGUET; BELLAICHE, 2017).

Esses ataques podem afetar não só o servidor alvo, mas também máquinas virtuais, recursos de rede e provedores, causando danos significativos e inviabilizando o serviço para usuários legítimos.

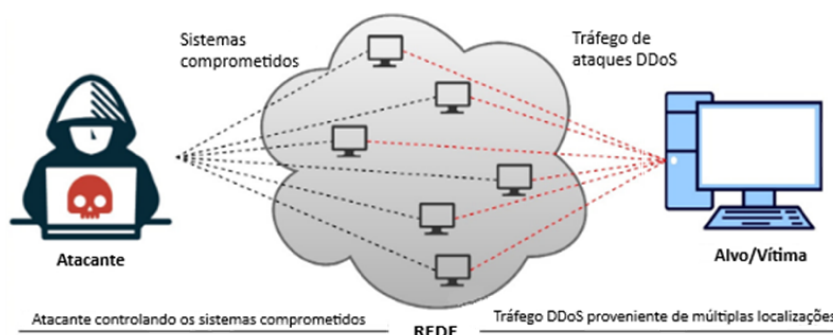


Figura 1 – Exemplo de funcionamento de um ataque DDoS. Adaptado de (KUMARI; JAIN, 2023).

2.1.2 Sistemas de detecção de intrusão (IDS)

Com o aumento das ameaças de segurança na rede, os sistemas de detecção de intrusão (IDS) passaram a ser amplamente estudados por apresentarem capacidade de monitorar, detectar e alertar sobre atividades suspeitas ou maliciosas em redes de computadores (LIAO et al., 2013).

Uma intrusão pode ser definida como qualquer tentativa de comprometer a confidencialidade, integridade ou disponibilidade de um sistema, bem como de burlar mecanismos de segurança de redes ou computadores. A detecção de intrusão consiste no monitoramento contínuo de eventos em um sistema ou rede, analisando-os em tempo real para identificar possíveis sinais de ataque. Um IDS pode ser implementado em software ou hardware, automatizando o processo de identificação e resposta a essas ameaças (LIAO et al., 2013).

Há dois tipos de IDS: os IDS baseados em *hosts* e os IDS baseados em rede (SINGH; SINGH, 2014). HIDS (*Host Based IDS*) é o primeiro tipo e sua principal função é monitorar internamente uma máquina e determinar se um sistema foi comprometido, mostrando avisos para os administradores. NIDS (*Network Based IDS*), por outro lado, monitoram a rede em si e analisam seu tráfego. Os NIDS leem todos os pacotes que chegam e caso algum possuir padrões suspeitos, emite um aviso para o administradores do sistema. NIDS também podem agir além de apenas emitir avisos, como ao barrar um endereço de IP de acessar a rede.

Há outras diferenciações entre IDS, como o IDS baseado em assinatura (SIDS) e IDS baseado em anomalia (AIDS). Em um SIDS, são usadas assinaturas (padrões ou *strings*) que correspondem com um ataque conhecido. O SIDS compara padrões com eventos capturados e reconhece possíveis intrusões. É um método simples e efetivo para detectar ataques já conhecidos, porém não é útil caso um ataque não estiver presente em seu banco de dados. Um IDS baseado em anomalia, entretanto, é efetivo para detectar ataques desconhecidos e novas vulnerabilidades. O AIDS é capaz de comparar perfis normais (aqueles que representam os comportamentos normais ou esperados derivados do monitoramento de atividades regulares, conexões de rede, *hosts* ou usuários durante um período de tempo) com eventos observados a fim de reconhecer possíveis ataques (LIAO et al., 2013).

Comparado aos SIDS, os AIDS podem apresentar uma probabilidade significativamente maior de gerar falsos positivos, especialmente quando a qualidade do conjunto de dados utilizado no treinamento é limitada ou quando há pouca semelhança entre ataques conhecidos e desconhecidos (KHRAISAT et al., 2019). Neste trabalho, serão explorados os IDS baseados em rede, com abordagem baseada na aplicação de técnicas de aprendizado de máquina para detecção de ataques, que vêm ganhando destaque por sua capacidade de identificar comportamentos desconhecidos ou variantes de ataques previamente observados (VISHWAKARMA; KESSWANI, 2023).

2.1.3 Aprendizado de máquina

O aprendizado de máquina (*Machine Learning*) é uma área da inteligência artificial dedicada ao desenvolvimento de algoritmos e modelos capazes de identificar padrões e executar tarefas com base em dados, sem a necessidade de programação explícita para cada situação específica. Por meio desse processo, é possível “ensinar” máquinas a processar e interpretar informações de forma mais eficiente, aprimorando seu desempenho com a experiência e a exposição a novos conjuntos de dados (MAHESH, 2020).

2.1.3.1 Aprendizado supervisionado e não supervisionado

O aprendizado de máquina pode ser classificado de acordo com a abordagem utilizada no processo de aprendizado, sendo as mais comuns o aprendizado supervisionado e não supervisionado. Essas abordagens diferem principalmente quanto ao uso de rótulos (valores de saída conhecidos para dados) durante o treinamento dos modelos (BI et al., 2019).

No aprendizado supervisionado, os valores dos resultados são conhecidos para cada observação (por exemplo, registros médicos destacados como câncer benigno ou câncer maligno) e são utilizados como rótulos para “treinar” o modelo. Esse tipo de dado, onde o valor do resultado é especificado, é conhecido como “dado rotulado”. Entre as técnicas de aprendizado supervisionado estão abordagens tradicionais, como regressão linear e regressão logística, além de algoritmos populares em aprendizado de máquina, como árvores de decisão e florestas aleatórias (OSISANWO et al., 2017).

No aprendizado não supervisionado, o algoritmo identifica padrões e agrupamentos nos dados sem utilizar valores conhecidos ou rótulos. Esse processo é semelhante a abordagens estatísticas que buscam subgrupos (*clusters*) com características semelhantes, como variáveis ou classes. Entre as técnicas, destacam-se o *k-means* e o agrupamento por maximização da expectativa (DAS; BEHERA, 2017).

Neste trabalho, os treinamentos dos modelos serão feitos com abordagens de aprendizagem supervisionada, em um cenário onde há apenas duas classes entre os dados, “Normal” para representar tráfego benigno e “DoS” para representar tráfego malicioso do tipo Negação de Serviço.

2.1.3.2 Conjuntos de dados

Para que o algoritmo seja capaz de identificar tais padrões, ele precisa ser treinado para poder fazer tais decisões. O treinamento de algoritmos de aprendizado de máquina é justamente os conjuntos de dados escolhido para o tal, que age como uma forma numérica de expressar experiências de um problema. Um processo de treinamento eficaz melhora a qualidade do sistema desenvolvido e a maneira com a qual esses dados são preparados influencia a performance do algoritmo. O processo deve ser realizado de forma cuidadosa, senão os resultados podem não ser bons o suficiente (UÇAR et al., 2020).

2.1.3.3 Treinamento e testes

Nesta etapa, os conjuntos de dados são divididos em duas partes: uma para treinamento e outra para teste. A proporção entre os dados de treinamento e de teste é um fator importante para determinar a taxa de sucesso do modelo: é recomendado que no mínimo o treinamento componha 50% do conjunto de dados (UÇAR et al., 2020).

O conjunto de treinamento é utilizado para que o algoritmo aprenda os padrões presentes nas amostras, ajustando seus parâmetros internos de acordo com o método de aprendizado escolhido. Durante essa fase, o modelo não tem acesso aos dados de teste, de forma a evitar a memorização de exemplos específicos e garantir o aprendizado.

Já o conjunto de teste é usado apenas após o treinamento, para avaliar o desempenho do modelo em uma parcela que não foi utilizada no treinamento. Essa separação é fundamental para medir a capacidade de generalização, evitando o problema de sobreajuste, cenário em que o modelo apresenta um desempenho excelente no treinamento, mas ao lidar com novos dados, sua eficácia decai.

Para aprimorar a fase de teste do modelo, também se utiliza a técnica de validação cruzada (*cross-validation*), na qual o conjunto de dados é dividido em múltiplas partes (*folds*), e o treinamento e a validação são repetidos diversas vezes, alternando os subconjuntos usados para cada etapa. Essa estratégia fornece uma melhor estimativa da performance do modelo.

2.1.4 Random Forest

Um dos algoritmos mais utilizados para classificação de ataques no contexto de IDS é a floresta aleatória (*Random Forest*) (RESENDE; DRUMMOND, 2018). O algoritmo é baseado no método de aprendizagem em conjunto (*ensemble learning*), que combina múltiplos modelos (no caso, árvores de decisão) para melhorar o desempenho preditivo e reduzir o risco de sobreajuste (BREIMAN, 2001).

Uma árvore de decisão é um modelo de aprendizado de máquina que funciona de forma semelhante a um conjunto de perguntas e respostas, organizadas em uma estrutura ramificada que lembra uma árvore. Cada nó interno da árvore representa uma decisão baseada em uma característica (variável) dos dados, e cada ramificação dessa decisão leva a outro nó ou a uma folha. As folhas correspondem às respostas finais do modelo e, em problemas de classificação como o abordado neste trabalho, indicam a atribuição a uma classe específica.

O RF então combina diversas destas árvores para gerar um resultado mais preciso. Cada árvore é treinada com um subconjunto aleatório do conjunto de dados, de forma a garantir diversidade entre as árvores geradas. O funcionamento geral do algoritmo é descrito a seguir:

- **Seleção de amostras com substituição (*bootstrap*):** O algoritmo cria várias amostras aleatórias do conjunto de dados original, permitindo repetição de exemplos em cada amostra.
- **Construção das árvores de decisão:** Para cada amostra, é construída uma árvore

de decisão, considerando apenas um subconjunto aleatório das variáveis (*features*) do conjunto de dados para decidir cada divisão.

- **Divisão recursiva dos dados:** Cada árvore divide seus dados de forma recursiva, escolhendo o melhor ponto de corte em cada nó, até atingir os critérios de parada, que podem ser a profundidade máxima ou número mínimo de exemplos por folha.
- **Repetição do processo:** O processo é repetido diversas vezes, gerando um conjunto (floresta) de árvores de decisão independentes.
- **Combinação das previsões:** Para problemas de classificação, cada árvore vota em uma classe, e a classe com mais votos é a predição final. Já para problemas de regressão, é calculado a média das previsões das árvores.
- **Resultado:** A combinação de várias árvores reduz o sobre-ajuste e aumenta a capacidade de generalização do modelo.

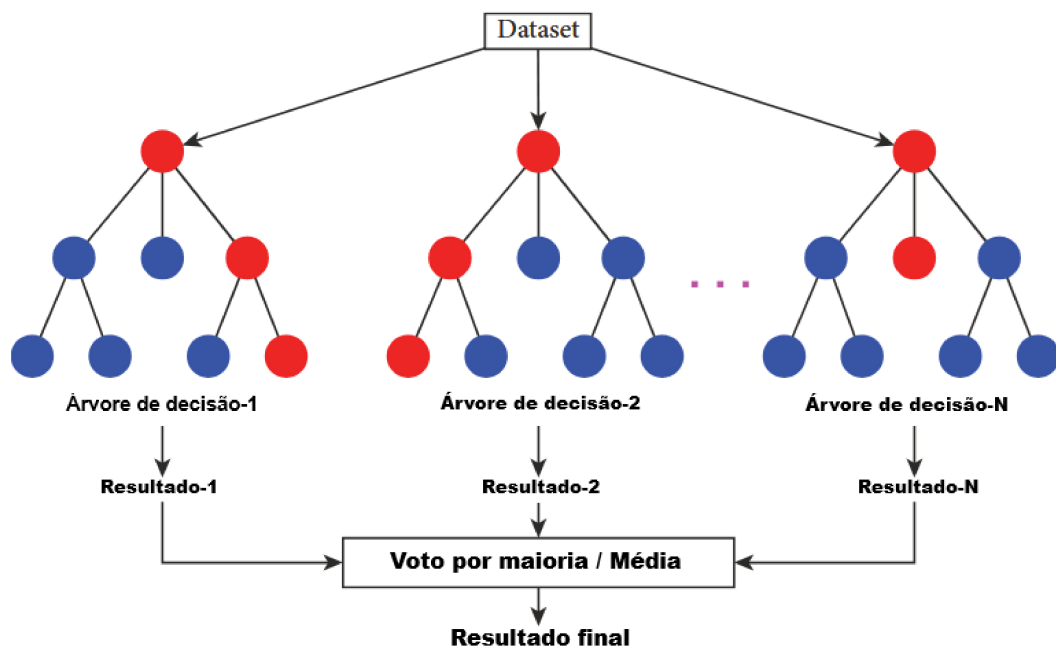


Figura 2 – Funcionamento geral do algoritmo Floresta Aleatória. Adaptado de (KHAN et al., 2021).

2.2 Trabalhos Relacionados

Em (SHARAFALDIN; LASHKARI; GHORBANI, 2018), os autores destacam a importância dos Sistemas de Detecção de Intrusão (IDS) na prevenção de ataques, especialmente diante do crescente impacto causado pela exposição de dados. Nesse contexto,

apontam que abordagens de IDS baseadas em anomalia enfrentam dificuldades devido à falta de conjuntos de dados adequados. Diversos *datasets* públicos apresentam problemas como baixa diversidade de tráfego e de ataques, desatualização, ausência de tráfego de redes reais, remoção de informações sensíveis e volume insuficiente de dados. Para superar essas limitações, o objetivo do trabalho foi criar um conjunto de dados que combinasse tráfego benigno e tráfego de ataques de diferentes tipos, de modo a refletir um cenário de mundo real. Para isso, os autores simularam duas redes: uma representando sistemas operacionais das vítimas e outra, os atacantes, executando múltiplos tipos de ataques. A partir dessa simulação, foi gerado o CICIDS2017. O artigo descreve detalhadamente a construção do conjunto de dados e será de fundamental importância para a utilização do mesmo no presente trabalho.

Em (SCHRÖTTER; NIEMANN; SCHNOR, 2024), os autores apresentam um estudo que analisa diversos trabalhos que propõem IDS baseados em redes neurais para a detecção de ataques desconhecidos em redes de Internet das Coisas (IOT). O estudo evidencia que grande parte dessas pesquisas não realiza comparações com IDS baseados em assinaturas ou com dados novos, o que dificulta a comprovação de avanços reais na área de detecção de intrusões. No artigo, replicam um modelo previamente proposto e o comparam com o *Snort IDS*, baseado em assinaturas, constatando um desempenho significativamente inferior ao apresentado no estudo original. Os erros observados consistiram principalmente na confusão em identificar os ataques ao invés de classificá-los como tráfego benigno.

O trabalho fornece percepções importantes para pesquisas futuras, como a necessidade de conjuntos de dados melhor estruturados, com maior variabilidade e a relevância em testar modelos com soluções estado da arte, como SIDS e com dados em ambientes distintos daqueles onde o modelo foi treinado. No presente estudo, essa preocupação será abordada ao testar modelos com dados gerados em cenários diferentes daqueles utilizados para a criação do conjunto de dados, neste caso, o CICIDS2017.

Em (ABDELAZIZ et al., 2024), é proposto o uso do algoritmo Floresta Aleatória aplicado ao conjunto de dados CICIDS2017 de forma a criar um NIDS com aprendizado de máquina. Os autores realizam pré-processamento e aplicam técnicas de seleção de variáveis para identificar as mais relevantes na detecção dos ataques. O treinamento do modelo resultou em um bom desempenho, com 99,8% na Medida-F1 ponderada e 93,31% na Medida-F1 macro em diferentes tipos de ataque. Esses resultados demonstram a capacidade do algoritmo em aprimorar a eficácia dos NIDS.

Em (D'HOOGHE et al., 2023), são destacadas as limitações recorrentes em artigos que exploram IDS baseados em aprendizado de máquina. É discutida a pouca investigação da capacidade de generalização de modelos além dos conjuntos de dados em que foram treinados. O autor então treina vários modelos com diversos algoritmos adotando o

dataset CSE-CIC-IDS2018 e os avaliam em outros conjuntos relacionados (CICIDS2017, CICDoS2017 e CICDDoS2019). Os resultados mostram que os modelos atingiram métricas elevadas no *dataset* de origem, mas não aprenderam representações aplicáveis a outros cenários, apresentando perdas de desempenho em várias classes de ataques, exceto em algumas como DoS, DDoS e *Brute Force*.

Um estudo mais recente, também sobre a generalização de modelos de NIDS é apresentado em (CANTONE; MARROCCO; BRIA, 2024). É investigada a capacidade de generalização de quatro classificadores, realizando a validação cruzada entre diferentes conjuntos de dados, incluindo o CICIDS2017, o CSE-CIC-IDS2018, o LycoS-IDS2017 e uma variante do último criada por eles. Os modelos foram avaliados com treinamentos feitos com seus próprios conjuntos de dados, obtendo uma performance quase perfeita. Porém, ao serem avaliados entre seus conjuntos de dados, o desempenho também caiu, como observado no último estudo, aproximando-se do acaso em algumas ocasiões.

Estes trabalhos apresentam resultados importantes em relação a generalização de modelos para IDS. Os autores concluem que o uso de apenas um *dataset* público não garante um desempenho necessário em aplicações reais, pois esses conjuntos acabam representando apenas instâncias específicas de ataques, relacionadas a ferramentas e parâmetros definidos no momento da captura. Fatores como erros ao rotular os dados, características do ambiente de coleta, topologia de rede e configurações de hardware influenciam a distribuição dos dados, comprometendo a generalização.

Diante dos trabalhos correlatos analisados, este estudo se diferencia por não se restringir apenas ao uso de conjuntos de dados públicos na avaliação dos modelos. Além do CICIDS2017, propõe-se a construção de um novo conjunto de dados, o UFUIDSDOS2025, elaborado em ambiente controlado com máquinas virtuais, no qual foram simulados os mesmos tipos de ataques presentes no CICIDS2017.

A partir disso, dois modelos são considerados: um treinado com o conjunto público e outro treinado com o conjunto criado neste trabalho. Assim, é feita uma investigação mais aprofundada sobre as diferenças de desempenho e, principalmente, sobre a capacidade de generalização entre ambos os modelos. Além desse tipo de comparação, este estudo também analisa as árvores de decisão de cada modelo, buscando identificar quais características de cada ambiente influenciam diretamente na generalização para amostras externas. A [Tabela 1](#) apresenta a contribuição de cada trabalho e a relação com a proposta deste.

Tabela 1 – Relação entre trabalhos correlatos e a proposta do trabalho. Fonte: Do Autor.

Trabalho	Contribuição principal	Relação com a proposta
(SHARAFALDIN; LASHKARI; GHORBANI, 2018)	Criação do conjunto de dados CICIDS2017, simulando tráfego benigno e malicioso em um ambiente controlado.	Fornece a base de dados principal utilizada.
(SCHRÖTTER; NIEMANN; SCHNOR, 2024)	Avaliação de IDS baseados em redes neurais em IoT; identificou falhas de comparação com modelos baseados em assinaturas e baixo desempenho fora do contexto original.	Mostra a importância de avaliar modelos em dados distintos para verificar generalização, foco deste trabalho.
(ABDELAZIZ et al., 2024)	Aplicação de Floresta Aleatória no CICIDS2017 com alta Acurácia e Medida-F1.	Mostra um bom desempenho interno no CICIDS2017 com o algoritmo Floresta Aleatória, que será utilizado neste trabalho.
(D'HOOGHE et al., 2023)	Testou modelos em diferentes <i>datasets</i> (CICIDS2017, CSE-CIC-IDS2018, etc.); observou queda significativa de desempenho fora do <i>dataset</i> de origem.	Evidencia os problemas de generalização em NIDS, motivando os experimentos relacionados a essa questão neste estudo.
(CANTONE; MARROCCO; BRIA, 2024)	Validou generalização de quatro classificadores em múltiplos conjuntos de dados (CICIDS2017, CSE-CIC-IDS2018, LycoS-IDS2017); desempenho caiu entre conjuntos.	Alinha-se diretamente com este trabalho, que também avalia a capacidade de generalização entre diferentes cenários.

3 Desenvolvimento

Neste capítulo, são apresentados todos os passos realizados para alcançar o objetivo proposto neste trabalho. As seguintes bibliotecas python foram utilizadas para as etapas:

- **Pandas**: Manipulação e análise dos dados.
- **Matplotlib** e **Seaborn**: Visualização dos dados.
- **Scikit-learn (Sklearn)**: Uso de algoritmos de aprendizado de máquina, métricas de avaliação e validação cruzada.
- **Imbalanced-Learn**: Ferramentas para lidar com desbalanceamento de classes como o SMOTE (*Synthetic Minority Oversampling Technique*).
- **Nfstream**: Biblioteca utilizada para extrair variáveis de rede do CICIDS2017 e criação do conjunto UFUIDSDOS2025.

3.1 Visão Geral

A Figura 3 ilustra a visão geral das etapas de desenvolvimento.

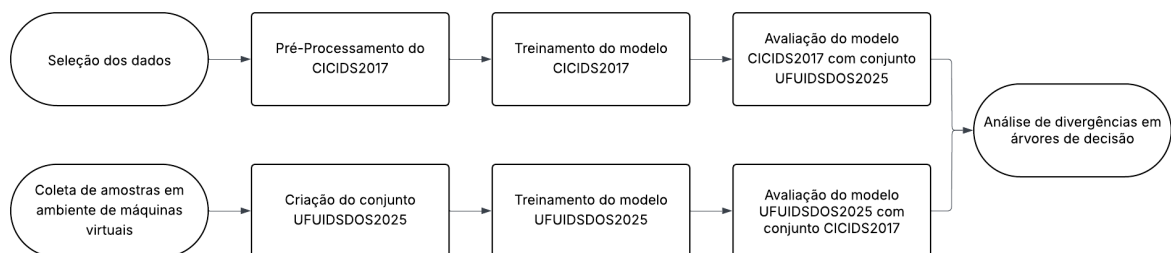


Figura 3 – Visão geral do desenvolvimento do trabalho. Fonte: Do Autor.

O trabalho foi dividido entre as seguintes seções:

- **Seleção dos dados**: Discussão dos dados públicos que foram escolhidos para serem utilizados para construção do primeiro modelo.
- **Pré-processamento do CICIDS2017**: Detalhamento de todas as técnicas de pré-processamento realizadas no conjunto CICIDS2017.

- **Treinamento do modelo do CICIDS2017:** Passos realizados para a criação do primeiro modelo, hiper-parâmetros utilizados e métricas de avaliação.
- **Criação do conjunto de dados UFUIDSDOS2025 e teste com o modelo CICIDS2017:** Detalhamento do processo de criação do conjunto de dados com amostras coletadas no ambiente de máquinas virtuais e passos seguidos para avaliação do primeiro modelo com esse conjunto.
- **Treinamento do modelo UFUIDSDOS2025 e teste com conjunto CICIDS2017:** Passos realizados para a criação do segundo modelo.
- **Análise de divergências em árvores de decisão:** Detalhamento das estratégias utilizadas para a análise das árvores de decisões do modelo.

3.2 Seleção dos dados

Para este trabalho, foi selecionado o conjunto de dados CICIDS2017 ([SHARAFAL-DIN; LASHKARI; GHORBANI, 2018](#)), criado pelo *Canadian Institute for Cybersecurity*. Este *dataset* inclui tráfego legítimo e ataques variados, como DoS, DDoS, *Brute-Force*, *Infiltration*, *Botnet*, *Port-Scanning* e *Web-Attacks*, distribuídos ao longo de cinco dias de captura. A escolha foi motivada por seu uso frequente na literatura ([SCHRÖTTER; NIE-MANN; SCHNOR, 2024](#)), por abranger múltiplos vetores de ataque e por disponibilizar informações estatísticas de fluxos de rede que permitem análises detalhadas.

Sua popularidade se deve à abrangência dos ataques simulados e à metodologia de coleta de dados, que buscou reproduzir condições reais de tráfego. Contudo, alguns estudos também apontam limitações, como desbalanceamento de classes e presença de fluxos inválidos, fatores que exigem cuidados no pré-processamento e na análise para evitar distorções nos resultados do modelo ([PANIGRAHI; BORAH, 2018](#)).

O CICIDS2017 foi construído utilizando um ambiente controlado, os dados foram coletados em uma rede física com duas seções: uma rede de vítimas, conectadas por *firewalls*, roteadores e *switches* e outra composta por máquinas atacantes. As máquinas da rede de vítimas possuíam sistemas operacionais *Windows*, *Linux* e *Macintosh*. Já a rede atacante incluía um roteador, um *switch* e 4 máquinas, com sistemas operacionais *Kali-Linux* e *Windows 8.1*.

Para a geração de tráfego benigno, foi utilizado um sistema chamado de *B-Profile*, criado pelo mesmo instituto, responsável por simular padrões naturais de tráfego (*HTTP*, *HTTPS*, *FTP*, *SSH*, e *e-mail*) baseado em perfis reais de 25 usuários. As coletas ocorreram ao longo de cinco dias, incluindo um dia dedicado exclusivamente ao tráfego benigno e os demais com os diversos tipos de ataques mencionados. O conjunto de dados possui no total sete classes de ataques, dentre elas 16 tipos específicos.

Para a utilização do conjunto de dados e, posteriormente, a aplicação do modelo treinado em testes com o conjunto de dados UFUIDSDOS2025, foi empregada a ferramenta *nfstream* para realizar a captura e a conversão de dados brutos, originalmente no formato *Packet Capture* (PCAP), em um arquivo no formato *Comma Separated Values* (CSV).

O *nfstream* é uma biblioteca em *Python* desenvolvida para capturar, processar e extrair métricas de tráfego de rede de forma estruturada, transformando pacotes brutos em fluxos com atributos prontos para análise. A escolha dessa ferramenta, em vez de utilizar diretamente os dados disponibilizados e capturados com o *CICFlowMeter* — ferramenta empregada pelos autores do CICIDS2017 — deve-se ao fato de que o conjunto de dados adicional foi gerado a partir da simulação de ataques em um ambiente de máquinas virtuais e extraído utilizando-a. Assim, para garantir consistência no processo de extração de variáveis, tanto o UFUIDSDOS2025 quanto o CICIDS2017 foram extraídos com o *nfstream*.

3.3 Pré-processamento do CICIDS2017

O pré-processamento de dados é fundamental na construção de modelos de AM de qualidade. Essa etapa é responsável por preparar e transformar os dados brutos em formatos adequados para análise e treinamento de modelos. Dados de baixa qualidade ou mal formatados podem comprometer o desempenho do modelo, levando a interpretações incorretas e resultados imprecisos. Em conjuntos de dados igual o CICIDS2017, que possui 1.845.604 amostras, é ainda mais importante, pois a grande quantidade de registros amplifica o impacto de ruídos, valores ausentes e dados desbalanceados.

Esse processo envolve a limpeza de inconsistências, tratamento de valores ausentes, normalização de variáveis, redução de dimensionalidade e seleção de atributos relevantes. Um pré-processamento bem executado aumenta a qualidade da informação disponível, reduz ruídos e melhora a capacidade de generalização do algoritmo.

3.3.1 Remoção de colunas com valores ausentes e linhas duplicadas

Algumas colunas do conjunto de dados possuíam um grande número de valores ausentes e foram removidas, pois poderiam impactar negativamente a performance do modelo. Além disso, foram eliminadas linhas duplicadas, já que também introduzem ruídos no conjunto de dados. Em relação a valores *Not a Number* (*NaN*), a análise concluiu que não haviam valores deste tipo, portanto não foi necessário retirar.

3.3.2 Variáveis categóricas

Algumas variáveis provenientes da extração do *nfstream* do tipo categóricas (não numéricas) eram irrelevantes para a criação e treinamento do modelo, por conter valores que não expressavam o comportamento da rede. Essas variáveis foram removidas com intuito de diminuir a dimensionalidade do conjunto de dados e evitar qualquer ruído no processo de treinamento do modelo.

3.3.3 Remoção de variáveis com risco de viés

Algumas variáveis presentes no conjunto de dados coletado pelo *nfstream* poderiam enviesar o modelo. Variáveis como *src_ip* e *dst_ip* poderiam identificar diretamente um atacante ou vítima, fazendo o modelo “decorar” os endereços de IP ao invés de aprender padrões de comportamento. Variáveis de tempo que identificam o horário exato do fluxo de rede poderiam também identificar um ataque DoS, portanto foram removidas do *dataset*. Portanto, optou-se por remover tais variáveis e manter apenas aquelas relacionadas ao tráfego e ao comportamento de rede.

3.3.4 Normalização

Como o conjunto de dados CICIDS2017 possui um grande número de variáveis numéricas com escalas muito diferentes, como tamanho de bytes com valores muito altos e variáveis de tempo em milissegundos ou segundos, torna-se necessário aplicar a normalização. Sem essa etapa, os algoritmos podem atribuir maior importância a variáveis com valores naturalmente maiores, desequilibrando o treinamento do modelo. Neste trabalho, foi utilizado o *StandardScaler* da biblioteca *sklearn*. Esse padronizador transforma os dados para que cada coluna apresente média igual a zero e desvio padrão igual a 1.

3.3.5 Remoção de classes

Com o objetivo de treinar e testar um modelo específico para detecção de ataques de negação de serviço, optou-se por excluir do conjunto de dados original todas as classes referentes a outros tipos de ataque. Além disso, adotou-se uma abordagem binária, classificando apenas entre tráfego normal e tráfego malicioso do tipo DoS/DDoS. Assim, todas as subclasses de ataques presentes no CICIDS2017 foram unificadas em uma única classe denominada “DoS”.

3.3.6 Seleção de variáveis

A seleção de variáveis é a fase do processamento responsável por escolher quais atributos do conjunto de dados realmente contribuem para o modelo. É possível eliminar

aquelas que são irrelevantes, redundantes ou que causam ruído. Isso pode ser feito com base em critérios estatísticos (como correlação), algoritmos de aprendizado (como importância de variáveis em modelos de árvore) ou heurísticas.

Neste trabalho, foram utilizados dois métodos para selecionar variáveis de forma a reduzir a dimensionalidade do conjunto de dados e melhorar a eficácia do modelo: Análise de correlação das variáveis e importância de variáveis com florestas aleatórias.

3.3.6.1 Análise de correlação

A análise de correlação é uma etapa do pré-processamento de dados que possui o objetivo de identificar relações estatísticas entre variáveis. Em conjunto de dados com muitas variáveis, como dados de tráfego de rede, diferentes atributos podem medir praticamente a mesma coisa: número total de pacotes, número de pacotes enviados ou tamanho total de bytes costumam estar fortemente correlacionados. Essas colunas redundantes não adicionam informação relevante, mas podem aumentar o tempo de treinamento, deixar o modelo mais complexo e até gerar vieses.

No conjunto de dados deste estudo, foi utilizada a correlação de *Pearson* para identificar o grau de relação entre duas variáveis. Variáveis com correlações próximas de +1 ou -1 apresentam comportamentos muito semelhantes. Quando isso acontece, é possível retirar uma das variáveis, que seria desnecessária para o modelo, de forma a melhorar seu desempenho. A seguir, é demonstrada a fórmula utilizada pelo *Pandas* para calcular a correlação entre duas variáveis:

$$r_{X,Y} = \frac{\text{cov}(X, Y)}{\sigma_X \cdot \sigma_Y}$$

Onde:

- $r_{X,Y}$: coeficiente de correlação de Pearson entre as variáveis X e Y ;
- $\text{cov}(X, Y)$: covariância entre X e Y , ou seja, como elas variam juntas;
- σ_X : desvio padrão de X , indicando a dispersão dos valores de X em torno da média;
- σ_Y : desvio padrão de Y , indicando a dispersão dos valores de Y em torno da média.

3.3.6.2 Importância de variáveis

A importância de variáveis é uma forma de saber quais atributos mais influenciam nas decisões de um modelo. Nas florestas aleatórias, esse processo é feito analisando o quanto cada variável contribui para reduzir a impureza das árvores usando *Gini* ou entropia. Neste trabalho, foi utilizado o *Gini*. O índice é utilizado para medir quão homogêneo

ou puro é um conjunto de dados em relação a uma determinada variável. Ele avalia a probabilidade de classificar incorretamente um elemento escolhido aleatoriamente nesse conjunto.

Um nó de uma árvore da floresta aleatória com alta impureza indica que as instâncias deste nó pertencem a classes diferentes com distribuições semelhantes, enquanto um nó com baixa impureza contém instâncias que pertencem de forma majoritária a uma única classe.

3.4 Treinamento do modelo do CICIDS2017

Neste estudo, o algoritmo escolhido para a criação do modelo de aprendizado de máquina foi o Floresta Aleatória, devido sua capacidade de classificação e seu extenso uso em pesquisas envolvendo a criação de IDS ([RESENDE; DRUMMOND, 2018](#)).

Para o processo de treinamento do modelo, primeiramente foi feita a divisão do conjunto de dados em dois subconjuntos de treino e teste. Essa divisão foi realizada com o método *train_test_split* da biblioteca *sklearn*, com o tamanho de teste especificado de 30%. Todos os dados dos subconjuntos de treinamento e teste foram normalizados, como mostrado na subseção de pré-processamento do conjunto de dados.

Primeiramente, foram escolhidos os hiper-parâmetros para o modelo, que foram ajustados para garantir uma melhor performance final. A seguir, estão cada um dos hiper-parâmetros utilizados, suas definições e os valores escolhidos:

- **n_estimators:** Representa o número de árvores da floresta. Quanto mais árvores, maior a capacidade do modelo, porém mais lento. Foram escolhidas 200 árvores para garantir diversidade suficiente para análise das estruturas das árvores.
- **max_depth:** É a profundidade máxima de cada árvore. Foi escolhido o valor 7, novamente motivado pela análise posterior de árvores do modelo e para evitar o sobre-ajuste ao conjunto de dados.
- **class_weight:** É responsável por dar mais peso às classes minoritárias para que o modelo não seja tendencioso em favor da classe majoritária. No cenário deste conjunto de dados, que possui um alto desbalanceamento de classes, foi optado por utilizar esse hiper-parâmetro como “*balanced*”, que ajusta automaticamente os pesos das classes inversamente proporcionais às suas frequências.

Para a avaliação do modelo, foram utilizadas métricas de desempenho como Acurácia, Precisão, Revocação e Medida-F1. Em suas fórmulas, são utilizadas as seguintes variáveis:

- **Verdadeiros Positivos (VP):** Quantidade de amostras que pertencem à classe analisada e foram corretamente classificadas como tal.
- **Falsos Positivos (FP):** Quantidade de amostras que pertencem à outra classe, mas foram incorretamente classificadas como à classe analisada.
- **Falsos Negativos (FN):** Quantidade de amostras que pertencem à classe analisada e foram incorretamente classificadas como à outra classe.
- **Verdadeiros Negativos (VN):** Quantidade de amostras da outra classe que foram corretamente classificadas como tal.

A partir dessas variáveis, são calculadas as seguintes métricas:

- **Acurácia:** Mede a proporção total de previsões corretas em relação ao total de classificações que o modelo realizou. Essa métrica pode ser enganosa em casos de conjunto de dados desbalanceados, portanto foi utilizada apenas como métrica complementar.

$$\frac{VP + VN}{VP + FP + FN + VN}$$

- **Precisão:** Mede quantos dos exemplos previstos como positivos realmente eram positivos.

$$\frac{VP}{VP + FP}$$

- **Revocação:** Mede quantos dos exemplos positivos reais foram devidamente identificados pelo modelo.

$$\frac{VP}{VP + FN}$$

- **Medida-F1:** Média harmônica entre precisão e revocação, é responsável por equilibrar as duas métricas, quando é desejável considerar tanto os falsos positivos quanto os falsos negativos, em cenários que há desbalanceamento.

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

Essas métricas foram apresentadas após o treinamento em formato “*Classification Report*”, realizado pelo *sklearn*. Além disso, foi utilizada a validação cruzada, responsável por avaliar o desempenho do modelo de forma mais confiável, dividindo o conjunto de dados em K partes chamadas *folds*. O modelo foi treinado várias vezes, cada vez usando um *fold* diferente como teste e o restante como treino.

Para avaliar com maior clareza a eficácia do modelo, utilizou-se também a matriz de confusão, que permite visualizar de forma detalhada as previsões corretas e incorretas em relação a cada classe.

3.5 Criação do conjunto de dados UFUIDSDOS2025 e teste com o modelo CICIDS2017

Nesta fase, foi desenvolvido um novo conjunto de dados para avaliar o desempenho do modelo recém treinado com o CICIDS2017. O objetivo foi verificar como o modelo se comportaria em um cenário distinto do ambiente original de coleta. Para isso, utilizou-se uma ferramenta de simulação em ambiente de rede com máquinas virtuais, capaz de reproduzir tráfegos de ataque, nos quais uma máquina A (atacante) realiza diferentes ataques contra uma máquina B (vítima). A ferramenta e suas instruções de uso estão disponíveis em [<https://github.com/gustavodgbernardo/ml-ids-attack-simulator>](https://github.com/gustavodgbernardo/ml-ids-attack-simulator)

Para a máquina atacante, adotou-se o sistema operacional *Kali Linux* 2024.3.1, por oferecer um alta gama de ferramentas voltadas para cibersegurança, testes de penetração e simulação de ataques. Já para a máquina vítima, adotou-se o sistema operacional *Ubuntu Linux* 16.04.7.

Para a geração do conjunto de dados UFUIDSDOS2025, utilizado na avaliação do modelo treinado com o CICIDS2017, foram realizados diferentes ataques de negação de serviço (DoS) contra a máquina vítima, especificamente dos tipos *DoS-GoldenEye*, *DoS-Hulk*, *DoS-Slowloris* e *DoS-SlowHttp*. Esses ataques já estavam presentes no treinamento do modelo do CICIDS2017, de forma que o objetivo principal foi verificar se este seria capaz de identificar corretamente instâncias da mesma classe de ataque em um cenário distinto. A seguir, a explicação de cada ferramenta utilizada para gerar os ataques:

- **DoS-GoldenEye**: Envia grandes quantidades de requisições HTTP usando conexões persistentes (*Keep-Alive*), para mantê-las abertas o máximo possível. Esgota o número de conexões disponíveis.
- **DoS-Hulk**: Envia um grande volume de requisições HTTP *GET*, cada uma modificada para dificultar bloqueio por filtros. Causa sobrecarga de CPU, memória e largura de banda.
- **DoS-SlowLoris**: Envia apenas os cabeçalhos HTTP de uma requisição de forma extremamente lenta para manter servidor esperando pela requisição completa, mantendo a conexão aberta.
- **SlowHttpTest**: Ferramenta de teste que permite simular vários tipos de ataques lentos (como o *Slowloris*).

Além dos ataques, também foi coletado tráfego benigno, garantindo a representação de ambas as classes. Utilizou-se a ferramenta *nfstream*, que capturou o tráfego de rede normal diretamente da interface de rede da máquina local, ao invés das máquinas

virtuais, durante um período de tempo. Os pacotes coletados foram organizados em fluxos bidirecionais e, a partir deles, foram extraídas estatísticas relevantes, posteriormente convertidas em um conjunto de dados no formato CSV.

O conjunto de dados resultante da simulação de ataques foi utilizado pelo modelo para fazer as previsões, permitindo avaliar sua capacidade de generalização em um ambiente distinto daquele do treinamento original. As instruções de obtenção do *dataset* UFUIDSDOS2025 estão disponíveis em [GitHub](#).

3.6 Treinamento do modelo UFUIDSDOS2025 e teste com conjunto CICIDS2017

Para possibilitar uma comparação mais consistente, foi construído um segundo modelo treinado com o UFUIDSDOS2025, posteriormente avaliado por meio de previsões no conjunto CICIDS2017. Este também foi treinado com o algoritmo Floresta Aleatória, utilizando os mesmos hiper-parâmetros adotados no treinamento do outro modelo.

A principal diferença nesse caso foi a aplicação do *SMOTE* (*Synthetic Minority Oversampling Technique*), para lidar com o desbalanceamento entre as classes, que neste caso, possuía poucas de tipo benigno. O *SMOTE* gera novas amostras sintéticas a partir das amostras existentes, desta forma, o conjunto de dados torna-se mais equilibrado, reduzindo o risco de sobre-ajuste, termo usado para descrever quando um modelo desempenha muito bem com dados do próprio conjunto treinado, porém se mostra ineficaz para prever novos resultados.

O objetivo da criação desse modelo foi realizar os mesmos testes de predição e análises das árvores de decisão aplicados ao modelo baseado no CICIDS2017, de forma a possibilitar uma comparação tanto em relação ao desempenho quanto à frequência de ocorrência de variáveis em nós de decisão de suas árvores.

3.7 Análise de divergências em árvores de decisão

Após a realização dos testes de predição com o primeiro modelo, treinado a partir do CICIDS2017, e com o segundo modelo, construído a partir do UFUIDSDOS2025, foram conduzidas análises com o objetivo de compreender o comportamento de ambos no processo de classificação das amostras em suas árvores de decisão. Para isso, utilizou-se a análise das árvores que compõem o algoritmo de Floresta Aleatória.

Foi desenvolvida uma função para rastrear os caminhos percorridos pelas amostras dentro das árvores de decisão e, ao identificar divergências nesses percursos, capturar o nó exato onde elas ocorrem. Como cada nó armazena informações sobre a variável utilizada na

divisão e o valor de limiar (*threshold*), tornou-se possível identificar quais variáveis exercem maior influência em decisões distintas entre previsões realizadas com o CICIDS2017 e com o UFUIDSDOS2025 em um ambiente diferente. O Algoritmo 1 representa a função para a captura das divergências, o qual retorna os dados de cada divergência em formato *dataframe* e o Algoritmo 2, a função auxiliar que rastreia o caminho de uma amostra.

Algoritmo 1: Função para detectar divergências entre caminhos de amostras de dois conjuntos em um modelo

Entrada: Modelo *rf_model*, conjuntos de amostras *X1* e *X2*, nomes *nome_X1*, *nome_X2*, lista de variáveis *feature_names*, número de amostras *n_amostras*

Saída: Tabela com divergências entre caminhos nas árvores da floresta

```

1 dados ← lista vazia;
2 Para cada tree_model em rf_model Faça
3     arvore ← tree_model.tree_;
4     Para i de 0 até n_amostras - 1 Faça
5         x1 ← amostra i de X1;
6         x2 ← amostra i de X2;
7         caminho1 ← caminho(arvore, x1);
8         caminho2 ← caminho(arvore, x2);
9         caminho_comum ← nós iguais iniciais de caminho1 e caminho2;
10        no_divergente ← último nó de caminho_comum;
11        Se no_divergente não for vazio Então
12            variavel ← variável do nó no_divergente;
13            limiar ← limiar do nó no_divergente;
14            profundidade ← comprimento de caminho_comum;
15            val_x1 ← valor da variável divergente de x1;
16            val_x2 ← valor da variável divergente de x2;
17            pred1 ← predição de x1 pelo tree_model;
18            pred2 ← predição de x2 pelo tree_model;
19            Se pred1 ≠ pred2 Então
20                adicionar linha com informações de divergência em dados;
21 Retorne dados;
```

Algoritmo 2: Função para rastrear o caminho de uma amostra na árvore de decisão

Entrada: Árvore de decisão *arvore*, amostra *x*
Saída: Caminho percorrido pela amostra na árvore

```

1 no ← 0;
2 caminho ← [no];
3 Enquanto arvore.children_left[no] ≠ arvore.children_right[no] Faça
4   |   variavel ← arvore.feature[no];
5   |   limiar ← arvore.threshold[no];
6   |   Se  $x[0, \textit{variavel}] \leq \textit{limiar}$  Então
7   |   |   no ← arvore.children_left[no];
8   |   Senão
9   |   |   no ← arvore.children_right[no];
10  |   caminho.append(no);
11 Retorne caminho;
```

Essa análise foi conduzida utilizando ambos os modelos treinados. Inicialmente, o modelo baseado no CICIDS2017 foi avaliado por meio da simulação dos caminhos de decisão com amostras de ataques DoS do próprio conjunto em que foi treinado, bem como com amostras de ataques DoS do UFUIDSDOS2025. Em seguida, o segundo modelo, treinado com os dados do ambiente de máquinas virtuais, foi submetido ao mesmo procedimento, sendo testado tanto com amostras do próprio conjunto quanto com amostras do CICIDS2017.

Foram coletadas informações como quais nós das árvores de cada modelo mais ocorreram divergências entre amostras de cada *dataset* e qual a distribuição dessas variáveis em ambos os conjuntos. A partir dessa análise, foi possível identificar as principais diferenças no comportamento das árvores de decisão de cada modelo e, conseqüentemente, extrair análises relevantes para a comparação entre os dois conjuntos de dados.

4 Resultados

Este capítulo apresenta os resultados obtidos a partir da execução das etapas descritas no desenvolvimento deste trabalho. Primeiramente, serão mostrados os resultados das técnicas aplicadas para o pré-processamento do conjunto de dados CICIDS2017. Após isso, serão discutidos os resultados referentes ao treinamento do modelo baseado no CICIDS2017 e aos testes de predição realizados com ele. Em seguida, será realizada a mesma análise do segundo modelo, treinado com o UFUIDSDOS2025, e seus testes com o CICIDS2017. Por fim, é apresentada a análise comparativa das árvores de decisão, extraídas dos dois modelos finais.

4.1 CICIDS2017 pós pré-processamento

Primeiramente, serão abordadas as mudanças do CICIDS2017 após as etapas de pré-processamento. O conjunto de dados original continha 1.845.604 amostras de tráfego de rede, incluindo registros benignos e de ataques, além de 88 variáveis extraídas pela ferramenta *nfstream*. As variáveis e seus respectivos significados estão disponíveis na seção *Nflow* da documentação oficial do *nfstream*: <<https://www.nfstream.org/docs/api#nflow-core-features>>

4.1.1 Limpeza do conjunto de dados e remoção de classes

Após a remoção de colunas com alto número de valores ausentes, linhas duplicadas e variáveis com risco de viés, o conjunto de dados passou a ter 1.626.568 amostras e 59 variáveis. A [Tabela 2](#) mostra as variáveis categóricas que foram removidas do conjunto de dados e seus respectivos significados. Já a [Tabela 3](#), mostra as variáveis de tempo que foram removidas. Essas poderiam indicar exatamente o momento que o ataque ocorreu em milissegundos e poderiam enviesar o modelo, portanto foi optado apenas por manter variáveis que indicam a duração do respectivo fluxo de rede.

A [Tabela 4](#) mostra a distribuição de classes anterior à remoção dos ataques desnecessários e a [Tabela 5](#) a distribuição após a remoção e binarização de tipos de ataques DoS e DDoS. Com a exclusão dos ataques não considerados neste estudo, foram eliminadas 9.675 amostras, o que corresponde a apenas 0,5934% do total.

Após a binarização dos rótulos do conjunto de dados, foi possível perceber mais claramente o desequilíbrio das classes do CICIDS2017, algo que foi levado em consideração na fase de treinamento do modelo.

Tabela 2 – Variáveis categóricas removidas. Fonte: Do Autor.

Variável	Significado
<i>src_ip</i>	Endereço IP de origem do fluxo
<i>dst_ip</i>	Endereço IP de destino do fluxo
<i>src_mac</i>	Endereço MAC de origem
<i>dst_mac</i>	Endereço MAC de destino
<i>src_oui</i>	Identificador do fabricante da placa de rede de origem (OUI)
<i>dst_oui</i>	Identificador do fabricante da placa de rede de destino (OUI)
<i>application_name</i>	Nome da aplicação identificada
<i>application_category_name</i>	Categoria da aplicação identificada
<i>requested_server_name</i>	Nome do servidor requisitado
<i>client_fingerprint</i>	Identificador do cliente
<i>server_fingerprint</i>	Identificador do servidor
<i>user_agent</i>	Informação do agente de usuário (HTTP)
<i>content_type</i>	Tipo de conteúdo transferido (HTTP <i>Content-Type</i>)

Tabela 3 – Variáveis de tempo removidas. Fonte: Do Autor.

Variável	Significado
<i>bidirectional_first_seen_ms</i>	Marca de tempo do primeiro pacote observado no fluxo bidirecional (milissegundos)
<i>bidirectional_last_seen_ms</i>	Marca de tempo do último pacote observado no fluxo bidirecional (milissegundos)
<i>src2dst_first_seen_ms</i>	Marca de tempo do primeiro pacote do fluxo de origem para destino (milissegundos)
<i>src2dst_last_seen_ms</i>	Marca de tempo do último pacote do fluxo de origem para destino (milissegundos)
<i>src2dst_duration_ms</i>	Duração total do fluxo de origem para destino (milissegundos)
<i>dst2src_first_seen_ms</i>	Marca de tempo do primeiro pacote do fluxo de destino para origem (milissegundos)
<i>dst2src_last_seen_ms</i>	Marca de tempo do último pacote do fluxo de destino para origem (milissegundos)

Tabela 4 – Distribuição das classes do conjunto de dados CICIDS2017. Fonte: Do Autor.

Classe	Número de ocorrências	Porcentagem
<i>Benign</i>	1.524.547	93,73%
<i>DDoS-LOIT</i>	62.736	3,86%
<i>DoS-Hulk</i>	14.072	0,87%
<i>DoS-GoldenEye</i>	7.547	0,46%
<i>DoS-Slowhttptest</i>	4.217	0,26%
<i>FTP-Patator</i>	3.972	0,24%
<i>DoS-slowloris</i>	3.774	0,23%
<i>SSH-Patator</i>	2.535	0,16%
<i>Web-Attack-Brute-Force</i>	1.353	0,08%
<i>Botnet-ARES</i>	741	0,05%
<i>Web-Attack-XSS</i>	676	0,04%
<i>Port-Scan</i>	343	0,02%
<i>Infiltration</i>	35	0,0022%
<i>Web-Attack-Sql-Injection</i>	12	0,0007%
<i>Heartbleed</i>	8	0,0005%

Tabela 5 – Distribuição das classes após remoção e binarização. Fonte: Do Autor.

Classe	Número de ocorrências	Porcentagem
Normal	1.524.547	94,29%
DoS	92.346	5,71%

4.1.2 Análise de correlação

Em seguida, realizou-se a análise de correlação entre as variáveis do conjunto de dados. Para facilitar a interpretação, utilizou-se um gráfico de mapa de calor da matriz de correlação, possibilitando uma melhor visualização das relações entre as variáveis, ilustrado na [Figura 4](#).

Com base na matriz de correlação, foram identificados os pares de variáveis com coeficiente absoluto superior a 0,85. Na sequência, contabilizaram-se as ocorrências de cada variável nesses pares altamente correlacionados e foi elaborado um gráfico de barras para melhor visualização das frequências. A partir dessa análise, as variáveis com maior recorrência foram removidas do conjunto de dados, com intuito de reduzir redundâncias. As variáveis eliminadas correspondem às 10 primeiras listadas no gráfico de maior recorrência em pares de alta correlação, apresentado na [Figura 5](#). Após a remoção, a dimensionalidade do conjunto de dados diminuiu para 1.616.893 amostras e 49 variáveis.

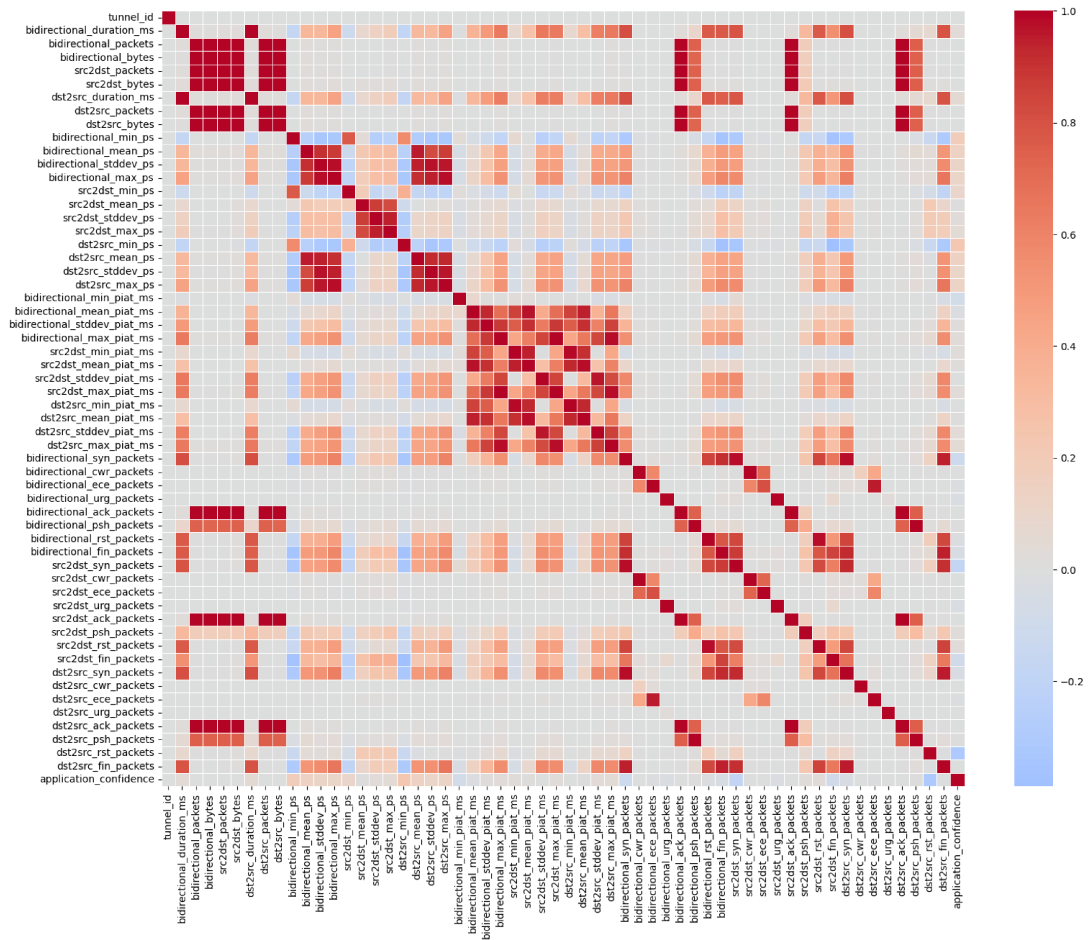


Figura 4 – Mapa de calor da matriz de correlação das variáveis do CICIDS2017. Fonte: Do Autor.

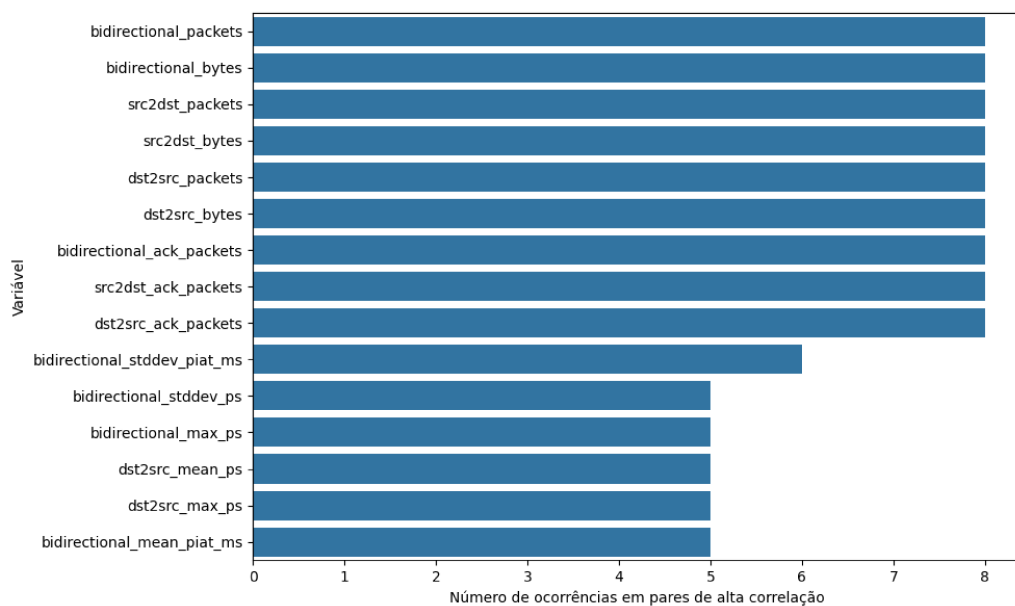


Figura 5 – Variáveis mais frequentes em pares de alta correlação do CICIDS2017. Fonte: Do Autor.

4.1.3 Análise de importância de variáveis

Em relação à importância de variáveis, foram coletadas as métricas de desempenho do modelo preliminar utilizando o algoritmo de Floresta Aleatória. A matriz de confusão correspondente é apresentada na Figura 6, enquanto os resultados da validação cruzada com 5 *folds* encontram-se na Tabela 6. Ambos os resultados apresentaram desempenho satisfatório, aumentando a confiabilidade da análise da importância relativa de cada variável nas decisões do modelo.

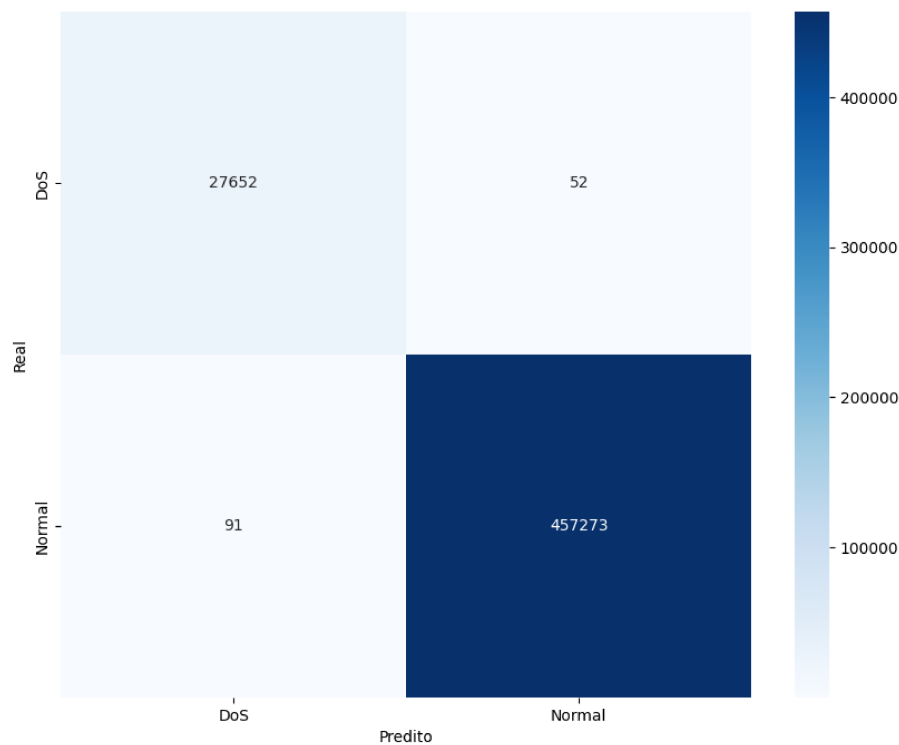


Figura 6 – Matriz de confusão do modelo de importância de variáveis. Fonte: Do Autor.

Tabela 6 – Resultados da validação cruzada do modelo para importância de variáveis. Fonte: Do Autor.

Fold	Acurácia
1	0.99975261
2	0.99969960
3	0.99973052
4	0.99969960
5	0.99971727

Foi feito um gráfico com os resultados do *feature_importances_* da biblioteca *sklearn* de modo a facilitar a visualização e a escolha de quais variáveis seriam removidas do conjunto de dados, a fim de diminuir a dimensionalidade e complexidade do CICIDS2017. As importâncias estão ilustradas na Figura 7.

A análise do gráfico de importância das variáveis revela que as primeiras 3 a 5 variáveis concentram a maior parte do poder preditivo do modelo, sendo principalmente métricas relacionadas ao tamanho e à variação dos pacotes, enquanto outras *features* apresentam pouca importância na predição. Essa concentração indica que o modelo depende fortemente dessas variáveis para distinguir entre tráfego normal e ataques DoS, o que será relevante para a análise das árvores de decisão posteriormente neste estudo.

Observa-se também que diversas variáveis apresentam importâncias significativamente baixas, de acordo com a distribuição decrescente mostrada no gráfico. No entanto, para evitar a perda de informações potencialmente úteis e garantir que o modelo mantenha um bom desempenho, optou-se por remover apenas um número limitado de variáveis, especificamente, as 11 com menor importância, ilustradas a partir da linha de corte no gráfico. O conjunto de dados passou a ter 1.616.893 amostras e 38 variáveis, marcando o fim das modificações realizadas no CICIDS2017.

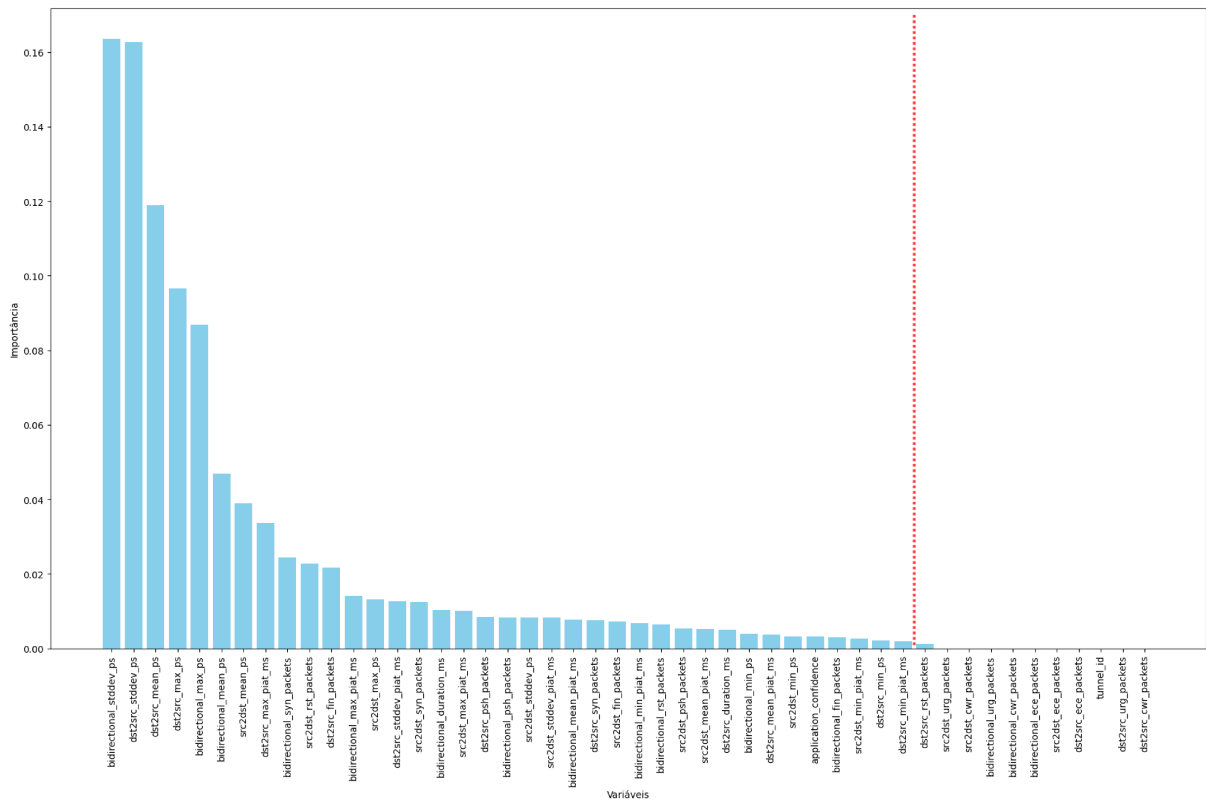


Figura 7 – Importância de variáveis utilizando florestas aleatórias. Fonte: Do Autor.

4.2 Treinamento do modelo do CICIDS2017

Após os resultados obtidos da limpeza do conjunto de dados, serão abordados os resultados do treinamento do modelo do CICIDS2017, utilizando o algoritmo Floresta Aleatória. O conjunto de testes contou com um total de 485.068 amostras, dentre elas

27.704 do tipo DoS e 457.374 do tipo Normal. Foi utilizado o *classification-report* da biblioteca *sklearn* para demonstrar valores como Precisão, Revocação e Medida-F1 do modelo. Além disso, é calculada a média macro (média simples das métricas) e média ponderada das métricas usando o suporte (quantidade de amostras) como peso.

A matriz de confusão é ilustrada na [Figura 8](#), o relatório de classificação do modelo é demonstrado na [Tabela 7](#) e os resultados dos *folds* da validação cruzada, na [Tabela 8](#).

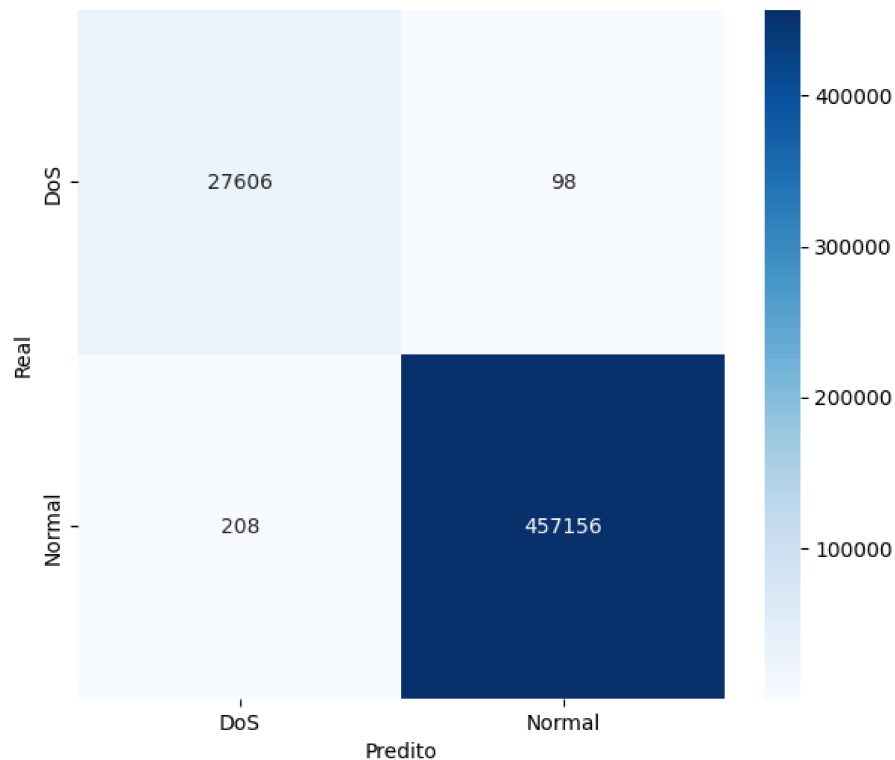


Figura 8 – Matriz de confusão do modelo do CICIDS2017. Fonte: Do Autor.

Tabela 7 – Relatório de classificação do modelo do CICIDS2017. Fonte: Do Autor.

Classe	Precisão	Revocação	Medida-F1	Suporte
DoS	0.99	1.00	0.99	27.704
Normal	1.00	1.00	1.00	457.364
Acurácia	1.00			485.068
Média macro	1.00	1.00	1.00	485.068
Média ponderada	1.00	1.00	1.00	485.068

Tabela 8 – Resultados da validação cruzada do modelo do CICIDS2017. Fonte: Do Autor.

Fold	Acurácia
1	0.9994743
2	0.99941687
3	0.99944779
4	0.99944779
5	0.99943012

O treinamento do modelo apresentou resultados satisfatórios, indicando que o algoritmo Floresta Aleatória conseguiu diferenciar de forma consistente o tráfego normal dos ataques DoS. Os valores de precisão, revocação e medida-F1 próximos de 1.00 mostram um desempenho quase perfeito. No entanto, esse tipo de resultado também pode sugerir a existência de sobre-ajuste (*overfitting*) ou refletir características específicas do conjunto de dados que facilitam a separação entre as classes, como variáveis específicas. Portanto, embora o modelo demonstre elevado desempenho com o CICIDS2017, é necessário avaliá-lo em outros cenários. No caso desse estudo, foi utilizados dados coletados em ambientes diferentes para avaliar sua capacidade de generalização.

4.3 Criação do conjunto de dados UFUIDSDOS2025 e teste com o modelo CICIDS2017

A coleta de fluxos de ataques do tipo *DoS-Hulk*, *DoS-Slowloris*, *DoS-SlowHttp* e *DoS-GoldenEye* e fluxos benignos resultou em um conjunto de dados com 49.703 amostras e 37 variáveis, dentre elas, 44.202 amostras do tipo DoS (com a binarização de ataques) e 5.501 amostras benignas.

A [Figura 9](#) mostra a distribuição de predições que o modelo do CICIDS2017 realizou. É possível perceber que o modelo não foi capaz de acertar as amostras do tipo DoS e demonstrou um desempenho baixíssimo. De 44.202 amostras, o modelo conseguiu prever apenas 5 da classe DoS, o que representa uma quantidade insignificante quando comparada com o total presente no UFUIDSDOS2025. Esse resultado evidencia uma limitação: apesar do desempenho quase perfeito no CICIDS2017, o modelo não conseguiu generalizar para um cenário diferente, falhando em identificar ataques nesse ambiente.

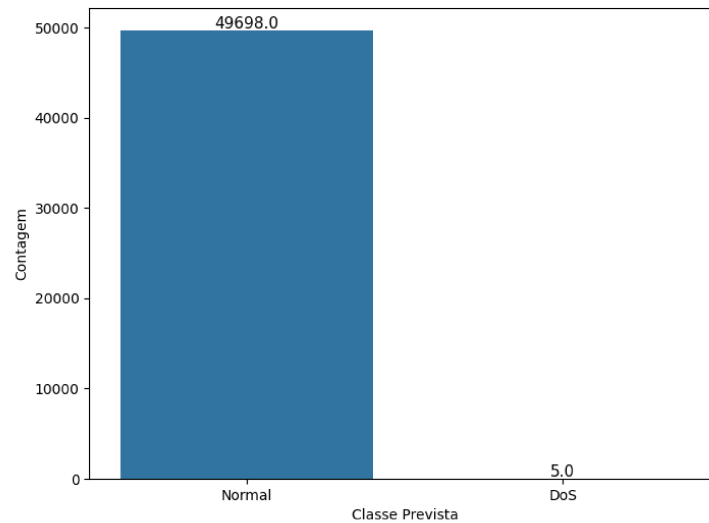


Figura 9 – Distribuição de previsões do modelo CICIDS2017 no UFUIDSDOS2025. Fonte: Do Autor.

A Figura 10 complementa a análise anterior. No eixo Y está representada a frequência de amostras, enquanto no eixo X consta a probabilidade atribuída pelo modelo de que determinada amostra pertença à classe DoS. Observa-se que a maior parte das amostras se concentra próxima de 0, indicando que o modelo apresentava alta confiança de que tais amostras não correspondiam a ataques DoS.

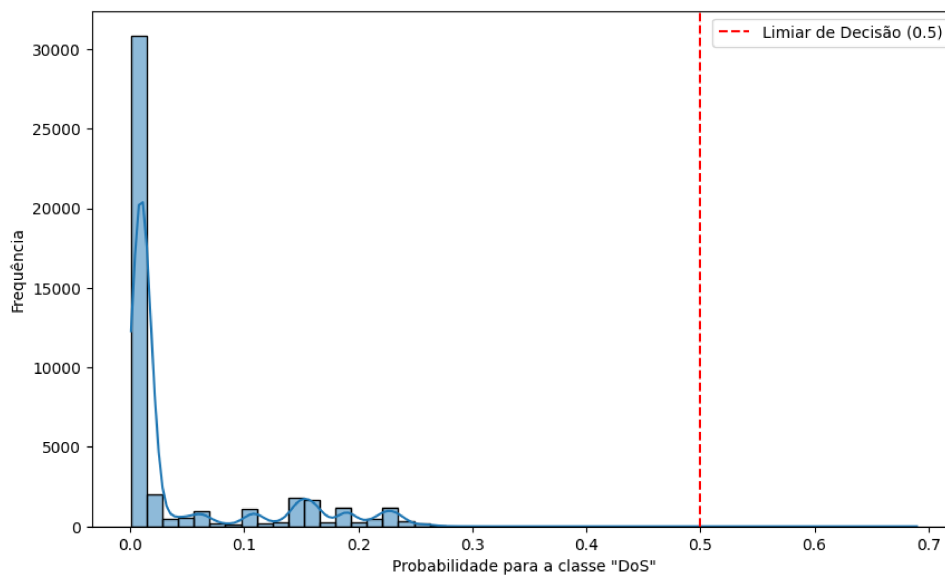


Figura 10 – Probabilidades de previsão para a classe DoS do modelo do CICIDS2017 no conjunto UFUIDSDOS2025. Fonte: Do Autor.

O limiar de decisão representado pela linha pontilhada vermelha demonstra um ponto em que há “dúvida” entre a classificação de uma amostra de uma classe ou outra. Neste caso, não houve nenhuma amostra próxima ou acima desse limiar, demonstrando

que o modelo não teve dúvidas ao classificar incorretamente os ataques como tráfego normal. Isso pode indicar que as características dos ataques no novo conjunto UFUIDSDOS2025 são tão diferentes daquelas vistas no treinamento que o modelo as considerou, com alta confiança, como pertencentes à classe “Normal”.

4.4 Treinamento do modelo do UFUIDSDOS2025 e teste com o CICIDS2017

Foram coletadas também as métricas de avaliação para o modelo treinado com o conjunto de dados de ataques DoS e tráfego benigno simulados. O conjunto de treino contou com 61.882 amostras no total, com 30.941 amostras do tipo DoS e a mesma quantidade para tráfego normal, por conta da sobre-amostragem realizada com o *SMOTE*. Já o conjunto de testes contou com 13.261 amostras do tipo DoS e 1.650 amostras benignas, totalizando 14.911.

Foram utilizadas as mesmas métricas e formas de apresentação dos resultados utilizados no treinamento do modelo treinado com o CICIDS2017. A [Tabela 9](#) e a [Tabela 10](#) apresentam, respectivamente, o resultado do relatório de classificação e a validação cruzada. Já a [Figura 11](#), ilustra a matriz de confusão das predições do modelo.

Tabela 9 – Relatório de classificação do modelo UFUIDSDOS2025. Fonte: Do Autor.

Classe	Precisão	Revocação	Medida-F1	Suporte
DoS	1.00	0.98	0.99	13261
Normal	0.87	1.00	0.93	1650
Acurácia			0.98	14911
Média Macro	0.94	0.99	0.96	14911
Média Ponderada	0.99	0.98	0.98	14911

Tabela 10 – Resultados da validação cruzada do modelo UFUIDSDOS2025. Fonte: Do Autor. Fonte: Do Autor.

Fold	Acurácia
1	0.99119334
2	0.98973903
3	0.99329347
4	0.99038462
5	0.9893342

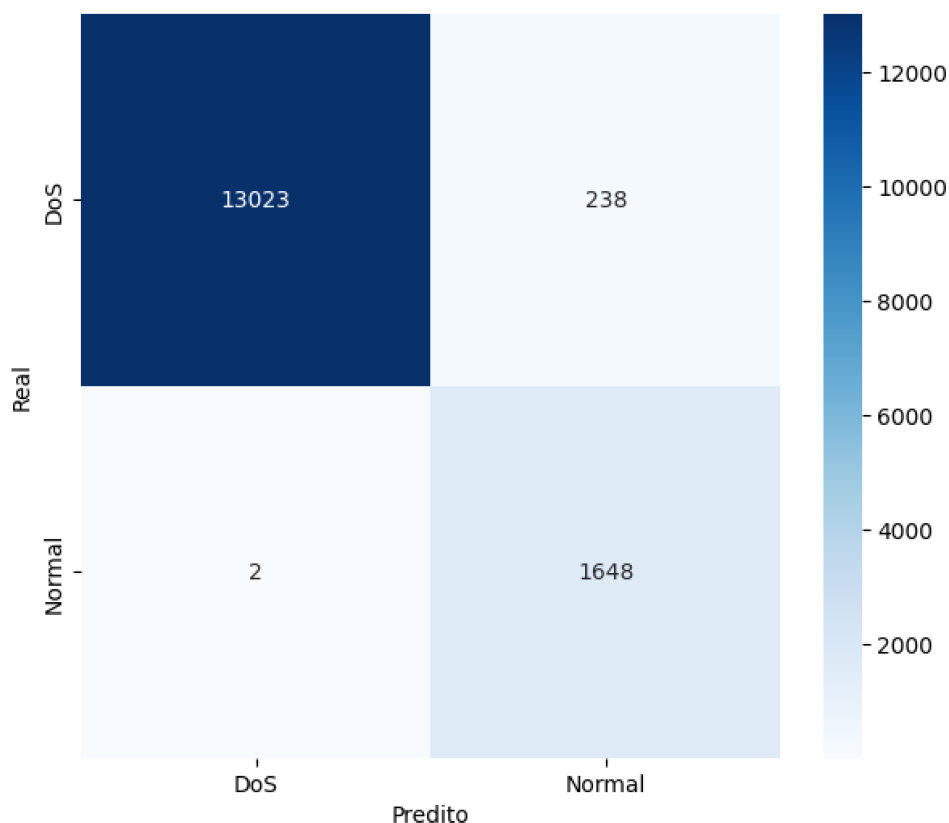


Figura 11 – Matriz de confusão do modelo UFUIDSDOS2025. Fonte: Do Autor.

Observa-se que o modelo treinado apresentou um desempenho satisfatório para a classe DoS. Entretanto, para a classe “Normal”, métricas como “Precisão” e “Medida-F1” apresentaram uma queda em comparação ao modelo baseado no CICIDS2017. Ainda assim, o desempenho geral pode ser considerado positivo, uma vez que o modelo manteve boa capacidade de identificação dos ataques. Essa redução de performance pode estar relacionada ao uso do *SMOTE* para gerar amostras sintéticas de fluxos benignos, o que pode ter introduzido distorções na representação dessa classe.

Nos testes de predição realizados com este modelo, o conjunto CICIDS2017 continha 1.524.547 amostras benignas e 92.346 amostras da classe DoS. Ao aplicar o modelo recém-treinado com o UFUIDSDOS2025, foram previstas 1.171.718 amostras como benignas e 445.175 como ataques DoS. Esse resultado evidencia uma capacidade relevante de generalização do modelo quando avaliado sobre o CICIDS2017 em comparação com o modelo anterior, que apresentou problemas de sobre-ajuste. A [Figura 12](#) ilustra seu gráfico de distribuição de predições.

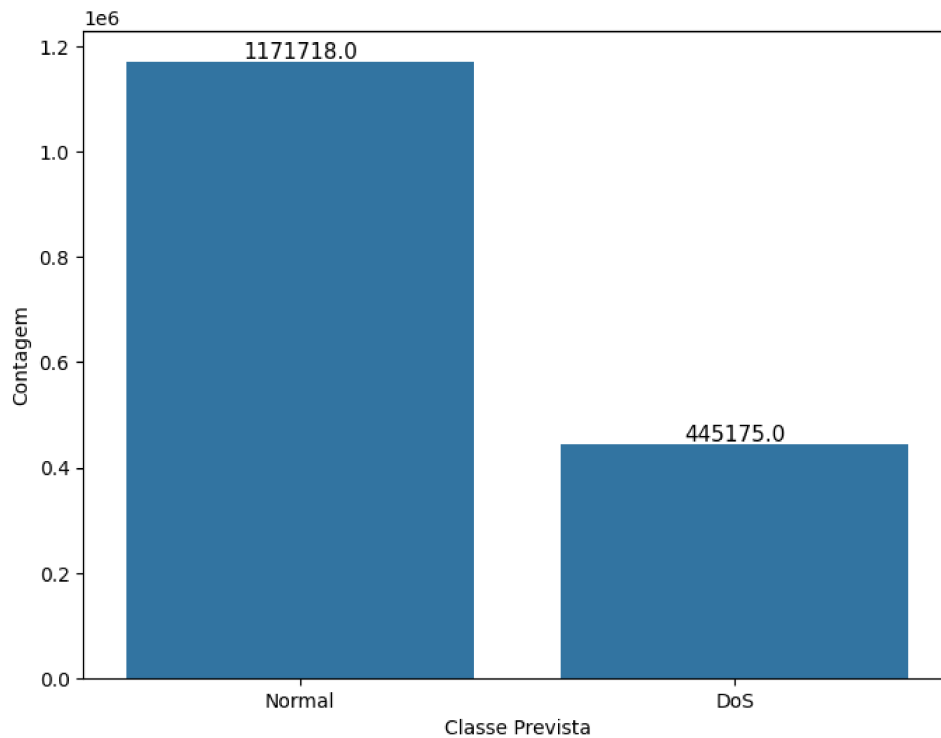


Figura 12 – Distribuição de previsões do modelo UFUIDSDOS2025 com amostras do CICIDS2017. Fonte: Do Autor.

4.5 Análise de divergências em árvores de decisão

Após a aplicação das funções para análise das divergências nas árvores do modelo treinado com o CICIDS2017, foram obtidas diversas informações sobre o comportamento dos grupos de amostras tanto do CICIDS2017 quanto do UFUIDSDOS2025. Em seguida, os mesmos testes foram conduzidos utilizando o modelo UFUIDSDOS2025, com o objetivo de comparar os resultados e identificar os principais pontos de divergência entre os modelos, no que se refere às decisões tomadas por suas árvores de decisão.

4.5.1 Divergências no modelo CICIDS2017

Primeiramente, foi feita uma análise dos pontos onde ocorreram divergências entre caminhos percorridos nas árvores por ambas as amostras que resultaram em uma divergência também na classe predita pelo modelo. A [Tabela 11](#) mostra as 10 primeiras amostras em que houveram divergências e ilustra o funcionamento da função. O significado de cada coluna da tabela é representado a seguir:

- **Amostra:** Número da amostra de ambos os conjuntos de dados.
- **Árvore:** Árvore da floresta onde ocorreu a divergência.

- **Prof.:** Profundidade da árvore em que ocorreu a divergência.
- **Variável Divergente:** Nome da variável responsável pela divergência.
- **Limiar:** Valor de separação de caminho entre as duas amostras.
- **Valor CIC:** Valor da variável na amostra do CICIDS2017.
- **Valor UFIIDS:** Valor da variável na amostra do conjunto UFIIDSDOS2025.
- **Classe CIC:** Em qual classe a amostra do CICIDS2017 chegou na árvore.
- **Classe UFIIDS:** Em qual classe a amostra do conjunto UFIIDSDOS2025 chegou na árvore.

Tabela 11 – Exemplo de divergências entre amostras do CICIDS2017 e conjunto UFIIDSDOS2025 no modelo do CICIDS2017. Fonte: Do Autor.

Amostra	Árvore	Prof.	Variável divergente	Limiar	Valor CIC	Valor UFIIDS	Classe CIC	Classe UFIIDS
0	0	1	bidirectional_max_ps	1.274	5.334	-0.264	DoS	Normal
1	0	1	bidirectional_max_ps	1.274	5.334	-0.226	DoS	Normal
2	0	1	bidirectional_max_ps	1.274	3.127	-0.306	DoS	Normal
3	0	1	bidirectional_max_ps	1.274	3.906	0.315	DoS	Normal
4	0	1	bidirectional_max_ps	1.274	3.906	-0.275	DoS	Normal
5	0	1	bidirectional_max_ps	1.274	3.906	-0.300	DoS	Normal
6	0	1	bidirectional_max_ps	1.274	2.417	-0.298	DoS	Normal
7	0	1	bidirectional_max_ps	1.274	3.906	-0.328	DoS	Normal
8	0	1	bidirectional_max_ps	1.274	3.906	-0.261	DoS	Normal
9	0	1	bidirectional_max_ps	1.274	3.875	-0.262	DoS	Normal

Foram realizados testes com um total de 1.000 amostras de cada conjunto de dados com a função de captura de divergências. A [Figura 13](#) exibe as variáveis presentes em nós de árvores em que mais ocorreram divergências entre amostras de ambos os conjuntos de dados. É possível perceber que as variáveis com as maiores frequências são variáveis que identificam tamanhos de pacotes dos fluxos de redes. Isso inclui o tamanho médio (*mean_ps*), o tamanho máximo (*max_ps*) e o desvio padrão (*stddev_ps*). Isso indica que os ataques do UFIIDSDOS2025 possuem perfis de tamanho de pacote que são drasticamente diferentes daqueles encontrados nos ataques do CICIDS2017. Outras variáveis que apareceram bastante foram as que representam *rst_packets*: pacotes responsáveis por encerrar imediatamente uma conexão TCP de forma abrupta.

Este comportamento pode explicar o motivo de amostras neste modelo serem preditas como “Normal” ao invés de ataque DoS, pois faz com que o modelo siga um caminho diferente na árvore de decisão e chegue a uma conclusão errada.

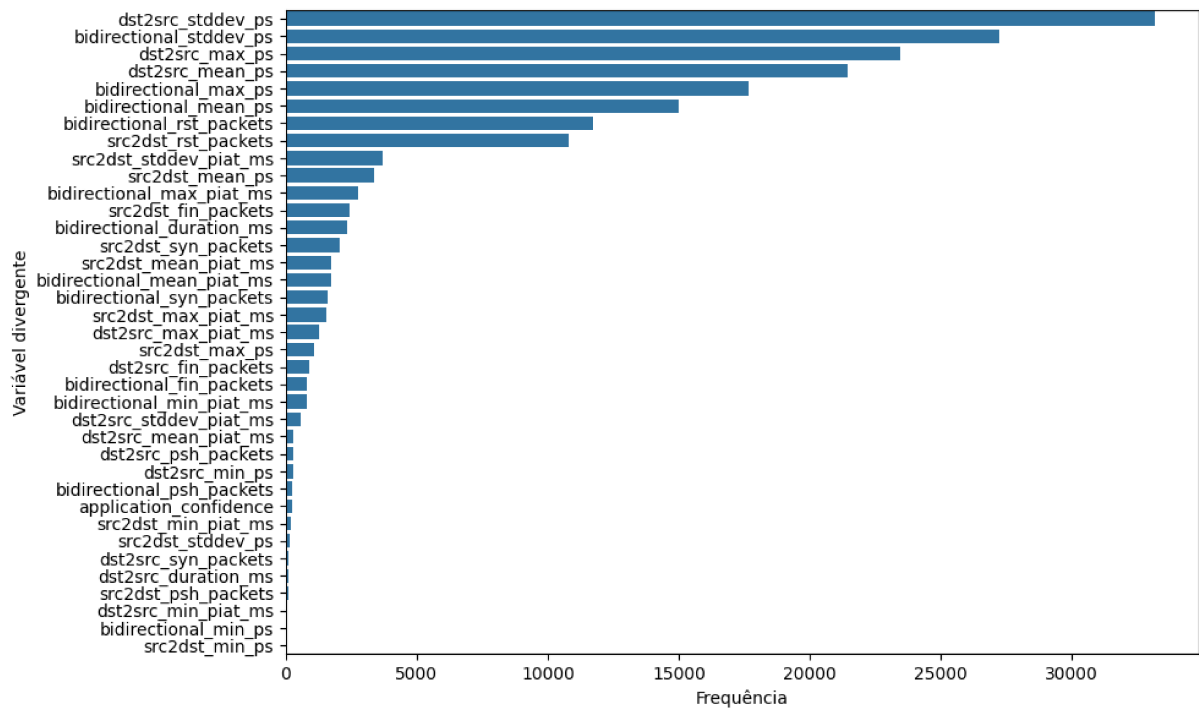


Figura 13 – Variáveis com mais divergências no modelo do CICIDS2017. Fonte: Do Autor.

A Figura 14 ilustra os *boxplots* da distribuição das variáveis mais divergentes em ambos os conjuntos de dados. É importante ressaltar que foram utilizados os respectivos conjuntos não normalizados, para melhor visualização das variações.

É possível observar que no CICIDS2017, várias variáveis, como *dst2src_max_ps* e *bidirectional_max_ps*, apresentam valores muito mais altos e uma dispersão muito maior, enquanto no UFUIDSDOS2025, os valores ficam concentrados em uma faixa menor.

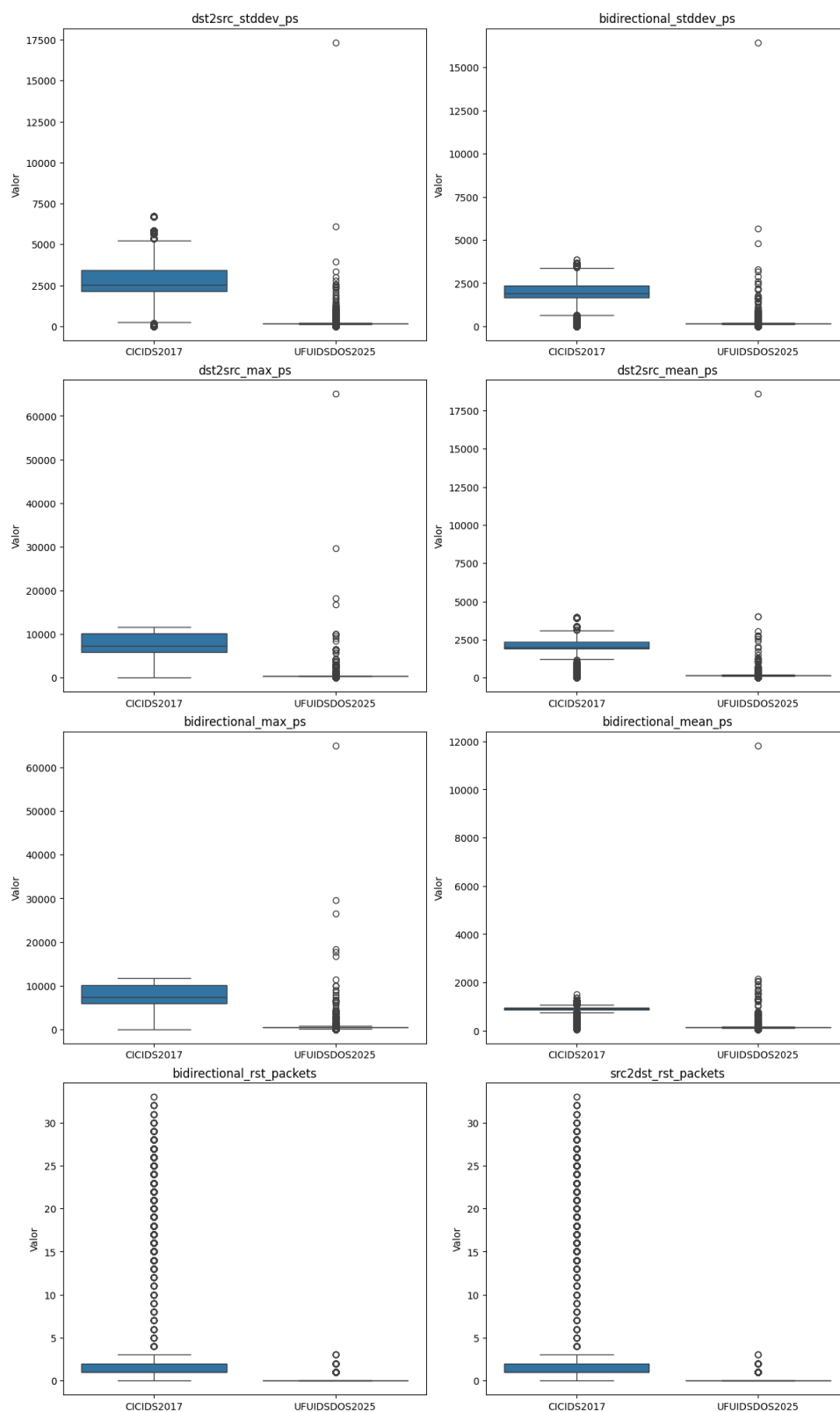


Figura 14 – *Boxplots* comparativos das variáveis mais divergentes no modelo CICIDS2017.
Fonte: Do Autor.

4.5.2 Divergências no modelo do UFUIDSDOS2025

Para os testes de divergência utilizando o modelo treinado com o UFUIDSDOS2025, foi selecionada a mesma quantidade de amostras de ambos os conjuntos de dados, para uma comparação similar. A distribuição das divergências observadas é apresentada na [Figura 15](#).

Neste cenário, verificou-se que variáveis relacionadas ao tamanho dos pacotes nos fluxos foram recorrentes nos nós divergente das árvores do modelo, destacando-se métricas como média, valor máximo e desvio padrão, do tipo “*bidirectional*” e “*dst_2_src*”, com maior ênfase nos fluxos no sentido destino a origem (vítima para atacante). Essas características são similares as variáveis mais divergentes apontadas na análise anterior, em relação ao modelo do CICIDS2017.

Além disso, observou-se divergências em variáveis associadas ao encerramento das conexões, como “*bidirectional_fin_packets*” e “*src2dst_fin_packets*”. Essas métricas correspondem ao número de pacotes com *flag FIN*, utilizados em finalizações de conexões de maneira normal. A relevância dessas variáveis indica que, no tráfego do UFUIDSDOS2025, o comportamento de encerramento das conexões de forma normal desempenha papel importante na diferenciação dos fluxos, diferente do que foi mostrado no modelo anterior, onde variáveis representando fluxos com pacotes com *flag RST* de encerramento abrupto possuíam mais divergências. Os *boxplots* comparativos destas variáveis são ilustrados na [Figura 16](#).

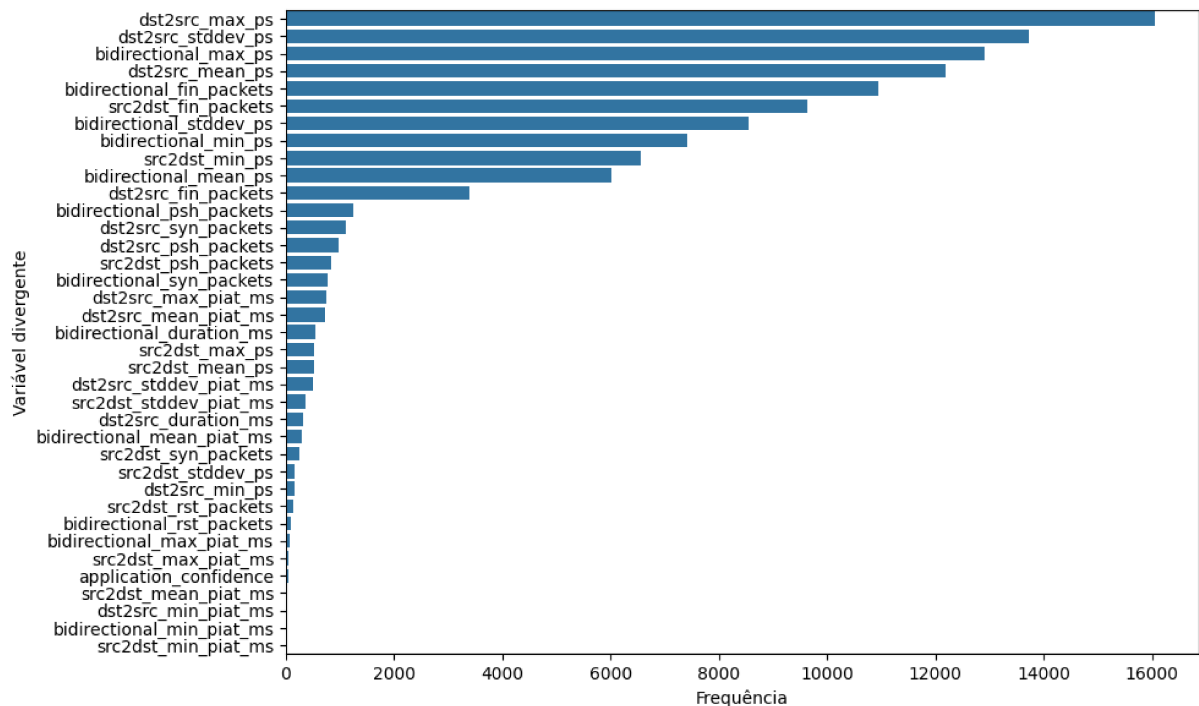


Figura 15 – Variáveis com mais divergências no modelo do UFUIDSDOS2025. Fonte: Do Autor.

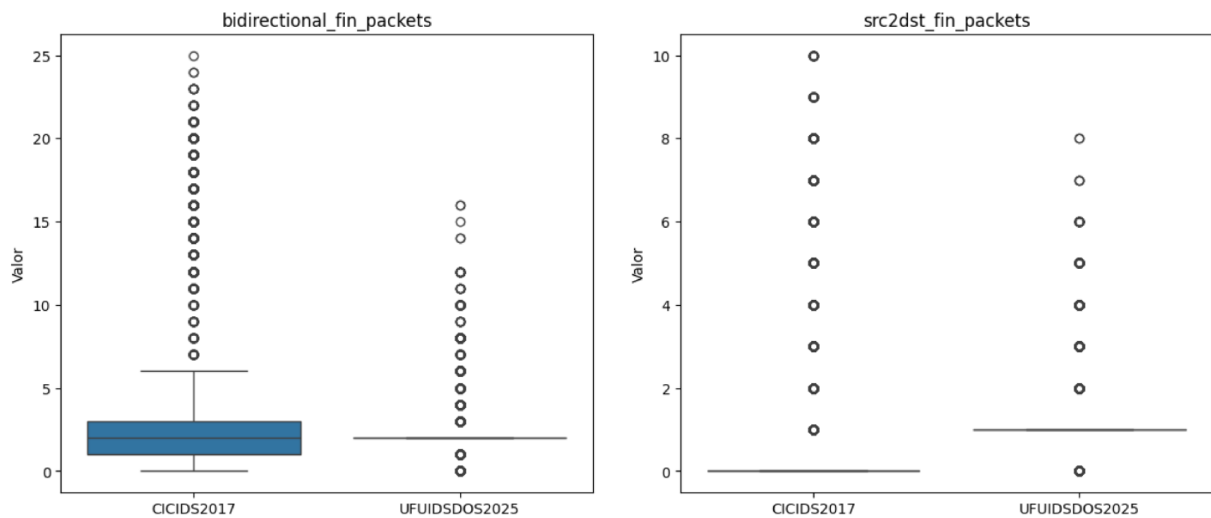


Figura 16 – *Boxplots* comparativos das variáveis de pacotes FIN. Fonte: Do Autor.

4.6 Discussão e Implicações Práticas

A análise do desempenho dos modelos treinados com diferentes conjuntos de dados e, posteriormente, a investigação das árvores de decisão do algoritmo Floresta Aleatória, revelou pontos importantes. Observou-se uma dificuldade de generalização — principal-

mente no modelo treinado com o CICIDS2017 — quando exposto a amostras de outro conjunto. Esse comportamento confirma uma sensibilidade dos modelos quando há variações no cenário de coleta.

A análise das árvores de decisão mostrou que certas variáveis, como as relacionadas ao volume de pacotes e bytes de tráfego, influenciaram na diferenciação das decisões. A forma como essas variáveis impactaram as predições foi diferente entre os conjuntos de teste: atributos que tiveram relevância em um ambiente não necessariamente tiveram a mesma importância em outro. Isso indica que o comportamento dos ataques pode variar de acordo com o cenário de coleta.

Os resultados deste capítulo se assemelham aos observados em (D’HOOGHE et al., 2023) e (CANTONE; MARROCCO; BRIA, 2024), que investigaram a capacidade de generalização de modelos no contexto de IDS. Assim como nesses estudos, verificou-se quedas significativas de desempenho quando modelos treinados em um conjunto de dados foram avaliados em outros, provenientes de ambientes distintos. Os conjuntos parecem capturar instâncias específicas de ataques e padrões de tráfego que não se repetem de forma similar em diferentes cenários, o que limita suas capacidades de generalização.

Do ponto de vista prático, esses resultados têm implicações para o desenvolvimento de Sistemas de Detecção de Intrusão baseados em aprendizado de máquina. Modelos treinados apenas em um *dataset* público, como o CICIDS2017, podem não ser eficazes quando adotados em ambientes reais, com padrões de tráfego distintos. Sugere-se, portanto, que ao desenvolver IDS baseados em aprendizado de máquina, seja explorada a possibilidade de utilizar outros conjuntos de dados, simulações ou tráfego real para realizar a avaliação do modelo.

5 Conclusão

Neste trabalho, foi proposta uma investigação do desempenho de um modelo treinado com o CICIDS2017 ao receber amostras coletadas em um ambiente distinto. Para isso, empregou-se o algoritmo Floresta Aleatória no treinamento de um modelo utilizando amostras do conjunto de dados CICIDS2017, com intuito de avaliar sua capacidade de generalização ao realizar previsões sobre fluxos de rede de ataques e de tráfego normal, coletados em outro ambiente. As amostras coletadas compuseram um conjunto de dados novo, denominado de “UFUIDSDOS2025”, o qual também foi utilizado para treinar um segundo modelo, permitindo a comparação de seu desempenho com o modelo inicial.

Para assegurar a qualidade dos dados em ambos os conjuntos e aprimorar o desempenho dos algoritmos, foram aplicadas técnicas de pré-processamento, como a limpeza dos dados e a remoção de classes e variáveis desnecessárias. Em seguida, analisou-se as capacidades de predição dos modelos e investigou-se as estruturas de suas árvores de decisão, de modo a possibilitar uma comparação detalhada entre os dois e entender suas principais diferenças.

A utilização do conjunto de dados UFUIDSDOS2025 para realizar previsões com o modelo treinado no CICIDS2017 evidenciou uma limitação na sua capacidade de generalização em relação aos ataques coletados no ambiente de máquinas virtuais. Mesmo tratando-se dos mesmos tipos de ataques DoS presentes no treinamento, somente 5 amostras foram corretamente classificadas como pertencentes à classe de ataques de negação de serviço, apesar do modelo original ter demonstrado um ótimo desempenho ao realizar testes com a parcela não utilizada no treinamento. Essa diferença indica a ocorrência de sobre-ajuste (*overfitting*): o modelo se ajustou muito bem às amostras específicas do CICIDS2017, mas não conseguiu manter o mesmo nível de desempenho ao lidar com novos conjuntos de dados, como o conjunto UFUIDSDOS2025 utilizado.

Em relação ao modelo treinado com o UFUIDSDOS2025, os resultados evidenciaram um desempenho inferior que o modelo do CICIDS2017 em relação ao conjunto separado de teste. Porém, ao empregar o modelo para realizar previsões no conjunto do CICIDS2017, diferente do passado, este conseguiu um resultado diferente: de 92.346 amostras originais, previu 445.175 como ataques DoS. Isso indica que o modelo mostrou uma maior sensibilidade aos ataques, mas perdeu em especificidade/precisão, ou seja, errou ao rotular amostras benignas como ataques. Esse resultado demonstra que, apesar de ter errado muitos rótulos benignos, o modelo conseguiu reconhecer ataques do CICIDS2017. Dito isso, é importante observar como um resultado interessante para o estudo, mas também como uma limitação de tal: O modelo possivelmente apresenta um viés em favor da

classe majoritária de ataque, influenciado pela sobre-amostragem realizada pelo *SMOTE*.

Por fim, a análise das árvores de decisão de ambos os modelos revelou que a principal diferença entre os ataques DoS do conjunto CICIDS2017 e aqueles provenientes do ambiente de máquinas virtuais está relacionada à distribuição do tamanho dos pacotes. As variáveis mais recorrentes nos pontos de divergência correspondem a métricas que descrevem, em sentido bidirecional e do destino para a origem, o desvio padrão (*stddev_ps*), o tamanho máximo (*max_ps*) e o tamanho médio (*mean_ps*) dos pacotes. Assim, enquanto o modelo treinado com o CICIDS2017 aprendeu a caracterizar ataques com base em um determinado padrão de tamanhos de pacotes, o modelo treinado no UFUIDSDOS2025 apresentou um padrão distinto. Essa discrepância explica a dificuldade do modelo treinado com o CICIDS2017 em generalizar de forma eficaz para as amostras do *dataset* UFUIDSDOS2025. Além disso, como variáveis divergentes secundárias, os modelos mostraram uma diferença em relação às *flags* de controle de conexão: o modelo CICIDS2017 apresentou divergências entre conjuntos para variáveis do tipo *RST* de encerramento abrupto de conexão, enquanto o modelo do UFUIDSDOS2025 apresentou divergências em relação às variáveis do tipo *FIN*, usados para terminar conexões de forma normal.

Como sugestão de trabalhos futuros, propõe-se a exploração aprofundada da interpretabilidade dos modelos por meio de técnicas como a biblioteca *python SHapley Additive exPlanations* (*SHAP*), a fim de identificar com maior precisão quais variáveis exercem maior influência nas classificações em diferentes cenários. Além disso, recomenda-se a investigação de outros algoritmos de aprendizado de máquina, incluindo abordagens baseadas em redes neurais profundas, de modo a comparar sua capacidade de generalização e desempenho em relação aos modelos treinados neste trabalho, seguindo a metodologia de aplicar testes utilizando outros conjuntos de dados, compostos por amostras diferentes daquelas utilizadas no treinamento.

Referências

- ABDELAZIZ, M. T.; RADWAN, A.; MAMDOUH, H.; SAAD, A. S.; ABUZOID, A. S.; ABDELHAKHEEM, A. A.; ZAKZOUK, S.; MOUSSA, K.; DARWEESH, M. S. Enhancing network threat detection with random forest-based nids and permutation feature importance. **Journal of Network and Systems Management**, Oct 2024. Citado 2 vezes nas páginas 19 e 21.
- BENDOVSCHI, A. Cyber-attacks – trends, patterns and security countermeasures. **Procedia Economics and Finance**, v. 28, p. 24–31, 2015. ISSN 2212-5671. 7th INTERNATIONAL CONFERENCE ON FINANCIAL CRIMINOLOGY 2015, 7th ICFC 2015, 13-14 April 2015, Wadham College, Oxford University, United Kingdom. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2212567115010771>>. Citado na página 13.
- BI, Q.; GOODMAN, K. E.; KAMINSKY, J.; LESSLER, J. What is machine learning? a primer for the epidemiologist. **American journal of epidemiology**, Oxford University Press, v. 188, n. 12, p. 2222–2239, 2019. Citado na página 16.
- BONGUET, A.; BELLAICHE, M. A survey of denial-of-service and distributed denial of service attacks and defenses in cloud computing. **Future Internet**, v. 9, n. 3, 2017. ISSN 1999-5903. Disponível em: <<https://www.mdpi.com/1999-5903/9/3/43>>. Citado na página 14.
- BREIMAN, L. Random forests. **Machine Learning**, v. 45, n. 1, p. 5–32, Oct 2001. ISSN 1573-0565. Disponível em: <<https://doi.org/10.1023/A:1010933404324>>. Citado na página 17.
- CANTONE, M.; MARROCCO, C.; BRIA, A. Machine learning in network intrusion detection: A cross-dataset generalization study. **IEEE Access**, v. 12, p. 144489–144508, 2024. Citado 3 vezes nas páginas 20, 21 e 50.
- DAS, K.; BEHERA, R. N. A survey on machine learning: concept, algorithms and applications. **International Journal of Innovative Research in Computer and Communication Engineering**, v. 5, n. 2, p. 1301–1309, 2017. Citado na página 16.
- DOULIGERIS, C.; MITROKOTSA, A. Ddos attacks and defense mechanisms: a classification. In: . [S.l.: s.n.], 2004. p. 190 – 193. ISBN 0-7803-8292-7. Citado na página 13.
- D’HOOGE, L.; VERKERKEN, M.; WAUTERS, T.; TURCK, F. D.; VOLCKAERT, B. Investigating generalized performance of data-constrained supervised machine learning models on novel, related samples in intrusion detection. **Sensors**, v. 23, n. 4, 2023. ISSN 1424-8220. Disponível em: <<https://www.mdpi.com/1424-8220/23/4/1846>>. Citado 3 vezes nas páginas 19, 21 e 50.
- KHAN, M. Y.; QAYOOM, A.; NIZAMI, M.; SIDDIQUI, M. S.; WASI, S.; SYED, K.-U.-R. R. Automated prediction of good dictionary examples (gdex): A comprehensive experiment with distant supervision, machine learning, and word embedding-based deep learning techniques. **Complexity**, 09 2021. Citado na página 18.

KHRAISAT, A.; GONDAL, I.; VAMPLEW, P.; KAMRUZZAMAN, J. Survey of intrusion detection systems: techniques, datasets and challenges. **Cybersecurity**, v. 2, n. 1, p. 20, Jul 2019. ISSN 2523-3246. Disponível em: <<https://doi.org/10.1186/s42400-019-0038-7>>. Citado 2 vezes nas páginas 11 e 15.

KUMARI, P.; JAIN, A. K. A comprehensive study of ddos attacks over iot network and their countermeasures. **Computers & Security**, v. 127, p. 103096, 2023. ISSN 0167-4048. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S0167404823000068>>. Citado na página 14.

LI, Y.; LIU, Q. A comprehensive review study of cyber-attacks and cyber security; emerging trends and recent developments. **Energy Reports**, v. 7, p. 8176–8186, 2021. ISSN 2352-4847. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2352484721007289>>. Citado na página 11.

LIAO, H.-J.; Richard Lin, C.-H.; LIN, Y.-C.; TUNG, K.-Y. Intrusion detection system: A comprehensive review. **Journal of Network and Computer Applications**, v. 36, n. 1, p. 16–24, 2013. ISSN 1084-8045. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S1084804512001944>>. Citado 2 vezes nas páginas 14 e 15.

MAHESH, B. Machine learning algorithms-a review. **International Journal of Science and Research (IJSR)**. [Internet], v. 9, n. 1, p. 381–386, 2020. Citado na página 15.

OSISANWO, F. Y.; AKINSOLA, J. E.; AWODELE, O.; HINMIKAIYE, J. O.; OLAKANMI, O.; AKINJOBI, J. et al. Supervised machine learning algorithms: classification and comparison. **International Journal of Computer Trends and Technology (IJCTT)**, v. 48, n. 3, p. 128–138, 2017. Citado na página 16.

PANIGRAHI, R.; BORAH, S. A detailed analysis of cicids2017 dataset for designing intrusion detection systems. **International Journal of Engineering & Technology**, v. 7, n. 3.24, p. 479–482, 2018. Citado na página 23.

RESENDE, P. A. A.; DRUMMOND, A. C. A survey of random forest based methods for intrusion detection systems. **ACM Comput. Surv.**, Association for Computing Machinery, New York, NY, USA, v. 51, n. 3, maio 2018. ISSN 0360-0300. Disponível em: <<https://doi.org/10.1145/3178582>>. Citado 2 vezes nas páginas 17 e 27.

SCHRÖTTER, M.; NIEMANN, A.; SCHNOR, B. A comparison of neural-network-based intrusion detection against signature-based detection in iot networks. **Information**, v. 15, n. 3, 2024. ISSN 2078-2489. Disponível em: <<https://www.mdpi.com/2078-2489/15/3/164>>. Citado 4 vezes nas páginas 12, 19, 21 e 23.

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 5 vezes nas páginas 11, 13, 18, 21 e 23.

SINGH, A. P.; SINGH, M. D. Analysis of host-based and network-based intrusion detection system. **International Journal of Computer Network and Information Security**, Modern Education and Computer Science Press, v. 6, n. 8, p. 41–47, 2014. Citado na página 15.

UÇAR, M. K.; NOUR, M.; SINDI, H.; POLAT, K. The effect of training and testing process on machine learning in biomedical datasets. **Mathematical Problems in Engineering**, Wiley Online Library, v. 2020, n. 1, p. 2836236, 2020. Citado na página 16.

VISHWAKARMA, M.; KESSWANI, N. A new two-phase intrusion detection system with naïve bayes machine learning for data classification and elliptic envelop method for anomaly detection. **Decision Analytics Journal**, v. 7, p. 100233, 2023. ISSN 2772-6622. Disponível em: <<https://www.sciencedirect.com/science/article/pii/S2772662223000735>>. Citado 2 vezes nas páginas 11 e 15.