Detecção de Ataques de Negação de Serviço Distribuídos com Algoritmos de Aprendizado de Máquina

Rodrigo R. Silva[1], **Felipe da R. Henriques**[1], **Igor M. Moraes** [2], **Dalbert M. Mascarenhas**[1]

[1]Centro Federal de Educação Tecnológica Celso Suckow da Fonseca - CEFET/RJ
Petrópolis - RJ - Brasil

[2]Laboratório MidiaCom – IC/TCC/PGC
Universidade Federal Fluminense (UFF), Niterói – RJ – Brasil

{felipe.henriques,dalbert.mascarenhas}@cefet-rj.br,
rodrigo.silva@aluno.cefet-rj.br, igor@ic.uff.br

Abstract. *This paper proposes a methodology to detect and classify distributed denial-of-service (DDoS) attacks. The proposed methodology employs data balancing techniques, preprocessing, and attribute selection that differ from those found in related work. We evaluate five machine learning algorithms, and we use the dataset CIC-DDoS2019 for training, validation, and evaluation. Experiments show that the Random Forest (RF) algorithm achieves the best results in both binary and multiclass classification. In the binary scenario without synthetic data, RF achieved 99.8% accuracy, while in multiclass classification, it reached a 100% detection rate for SYN attacks and 98% or higher for other types of attacks.*

Resumo. *Este artigo propõe uma metodologia para detectar e classificar ataques de negação de serviço distribuídos. A metodologia proposta emprega técnicas de balanceamento de dados, pré-processamento e seleção de atributos diferentes das encontradas nos trabalhos relacionados. São avaliados cinco algoritmos de aprendizado de máquina e o conjunto de dados usado para treinamento, validação e avaliação é o CIC-DDoS2019. Experimentos mostram que o algoritmo* Random Forest *(RF) apresenta os melhores resultados tanto na classificação binária quanto na classificação multiclasse. No cenário binário sem dados sintéticos, o RF atingiu 99,8% de acurácia, enquanto na classificação multiclasse alcançou uma taxa de detecção de 100% para ataques SYN e 98% ou superior para outros tipos de ataques.*

1. Introdução

Nos ataques de negação de serviço distribuídos (*Distributed Denial of Service* - DDoS) inúmeros sistemas finais agem com o objetivo comum de tornar inacessíveis os serviços providos pela vítima a usuários legítimos [Laufer et al., 2005]. Uma abordagem simples para atingir tal objetivo é inundar a vítima com um grande número de mensagens de forma a consumir quase que a totalidade dos seus recursos [Bala e Behal, 2024]. Os ataques DDoS são uma ameaça mesmo para vítimas superdimensionadas. Sempre é possível inundar uma vítima e negar o seu serviço desde que um número suficiente de sistemas finais participe do ataque. Em fevereiro de 2022, ocorreu um dos maiores ataques DDoS já registrados. Foi usada uma rede de *bots* (*botnet*) composta por mais de 30

mil *bots*, resultando em mais de 71 milhões de requisições HTTP/2 por segundo. Essa taxa foi 54% superior à taxa recorde anterior de 46 milhões de requisições por segundo, registrada em junho de 2022 [Yoachimik et al., 2023]. Esses incidentes ressaltam a crescente sofisticação dos ciberataques e a importância de medidas para detecção de ataques DDoS [Li et al., 2023, Lima et al., 2023, Agiollo et al., 2023, Horchulhack et al., 2022].

O uso de algoritmos de aprendizado de máquina é uma alternativa eficiente para identificar e classificar os ataques DDoS [Polat et al., 2020, Nazarudeen e Sundar, 2022, Kurniabudi et al., 2020, Elsayed et al., 2020, Arp et al., 2022]. Esses algoritmos identificam padrões sutis e complexos nos dados, difíceis de detectar por métodos tradicionais de detecção e classificação ou observação humana. A capacidade continua de aprendizagem e adaptação a novos cenários faz os algoritmos de aprendizado de máquina serem indicados para análises, principalmente, em que há variabilidade de padrões de tráfego associados a diferentes tipos de ataques. Este artigo propõe uma metodologia para detectar e classificar ataques DDoS que usa algoritmos de aprendizado de máquina. Com a metodologia proposta, avaliam-se cinco algoritmos de aprendizado de máquina para detecção de ataques e classificação dos tipos de ataques DDoS presentes no conjunto de dados (*dataset*) CIC-DDoS2019 [Sharafaldin et al., 2019]. São eles: *Naive Bayes* (NB), *MultiLayer Perceptron* (MLP), Árvore de Decisão (*Decision Tree* - DT), *Random Forest* (RF) e *Support Vector Machine* (SVM). O objetivo é investigar os algoritmos que melhor se adaptam ao contexto da detecção e classificação de ataques DDoS, em termos da acurácia, precisão, *recall*, F1-*score* e ROC-AUC (*Area Under the* ROC *Curve*). A metodologia proposta emprega técnicas de balanceamento de dados, pré-processamento e seleção de atributos diferentes das encontradas nos trabalhos relacionados. Os diferenciais da metodologia proposta incluem a possibilidade de balanceamento do conjunto de dados com dados sintéticos através da técnica *Synthetic Minority Over-sampling Technique* (SMOTE), que cria novas amostras de dados com base nas amostras pré-existentes, e uma etapa de pré-processamento de dados, na qual não são considerados atributos *socket* específicos de rede e que indicam a entrada ou saída do fluxo de rede, como o atributo *"Inbound"*. A seleção de atributos é realizada em duas etapas: a primeira combina duas técnicas de seleção, sendo limite de variância e o método *feature_importances_* do algoritmo *Random Forest*, enquanto a segunda etapa envolve uma análise manual da matriz de correlação que aplica um filtro aos atributos selecionados na primeira etapa. Além disso, são compartilhados os hiperparâmetros dos modelos implementados após extensivos testes manuais anotando cada resultado para garantir, após análises, um aumento de desempenho. O trabalho se concentra exclusivamente na seleção, implementação e comparação dos métodos de aprendizado, considerando sua aplicabilidade e desempenho na tarefa de detecção em diferentes cenários propostos com e sem geração de dados sintéticos e na classificação do tipo de ataque. No cenário binário reduzido e sem dados sintéticos, o RF atingiu 99,8% de acurácia. Na classificação multiclasse, o RF alcançou 100% de taxa de detecção para ataques SYN e 98% ou acima para os demais ataques analisados, com enfoque em sete tipos de ataques. Há indícios também de que a metodologia proposta supera outras metodologias propostas na literatura [Nazarudeen e Sundar, 2022, Elsayed et al., 2020] no cenário binário com dados sintéticos em termo de precisão e acurácia.

O restante deste artigo está organizado da seguinte forma. A Seção 2 discute os trabalhos relacionados. A Seção 3 explica a metodologia proposta. A Seção 4 discute os resultados obtidos. A Seção 5 conclui este trabalho.

2. Trabalhos Relacionados

Diferentes trabalhos da literatura usam algoritmos de aprendizado de máquina para detectar e identificar ataques DDoS e empregam diferentes metodologias e diferentes conjuntos de dados.

Polat *et al*. usam algoritmos de aprendizado de máquina para detectar ataques DDoS em redes definidas por software (*Software Defined Networks* - SDN) [Polat et al., 2020]. O conjunto de dados usado no trabalho possui características específicas do SDN em condições normais e sob tráfego de ataque DDoS. Os autores criam um novo conjunto de dados usando um método de seleção de atributos com objetivo de simplificar os modelos, facilitar a interpretação e reduzir o tempo de treinamento. A seleção de atributos é feita com o *wrapper*, que é uma estratégia de busca para identificar possíveis subconjuntos de atributos. O processo de busca é repetido até que os atributos ideais sejam obtidos, o que é computacionalmente exigente devido à grande variedade de combinações de atributos possíveis. Ambos os conjuntos de dados, criados com e sem métodos de seleção de atributos, são treinados e testados com os algoritmos SVM, NB, Rede Neural Artificial (RNA) e *K-Nearest Neighbors* (KNN). Os resultados mostram que o uso da seleção de atributos do *wrapper* com o classificador KNN alcança a maior taxa de acurácia, 98,3%, na detecção de ataques DDoS.

Elsayed *et al.* também lidam com a detecção de ataques DDoS em SDN [Elsayed et al., 2020], porém usam o conjunto de dados CIC-DDoS2019, que não é específico para SDN. Segundo os autores, tal *dataset* abrange uma ampla gama de ataques DDoS preenchendo lacunas de conjuntos de dados anteriores. Os autores apresentam um sistema de detecção denominado DDoSNet que emprega *Deep Learning* e combina técnicas de Rede Neural Recorrente (RNN). Na etapa de pré-processamento, os autores removem do *dataset* tuplas com valores NaN e infinitos, usam a escala de dados [0:1] e codificação 0 e 1. Os resultados mostram que o sistema proposto pelos autores obtém o melhor desempenho quando comparado aos algoritmos DT, NB, *Booster*, RF, SVM e *Logistic Regression* (LR) com acurácia de 99% e pontuação ROC-AUC de 98,8%.

Kurniabudi *et al.* avaliam diferentes algoritmos de aprendizado de máquina para detecção de DDoS, em termos da precisão da detecção e do tempo de execução dos algoritmos [Kurniabudi et al., 2020]. O conjunto de dados usado é o CIC-IDS2017, que contém tráfego de rede normal e tráfego de diferentes tipos de ciberataques, incluindo DoS, DDoS, ataques de força bruta, escaneamento de portas, entre outros. Os autores não especificam a etapa de pré-processamento e selecionam os atributos de acordo com valor mínimo de peso. Os resultados mostram que o algoritmo *Random Forest* obtém a maior acurácia, alcançando 99,86% com 22 atributos relevantes, enquanto o J48 alcançou 99,87% com 52 atributos, porém, com maior tempo de execução.

Diferente dos trabalhos anteriores, Nazarudeen e Sundar apresentam uma abordagem de detecção e também de identificação de ataques DDoS, que limita o espaço de atributos para minimizar o sobreajuste e tempo computacional dos modelos propostos [Nazarudeen e Sundar, 2022]. Os autores usam o conjunto de dados CIC-DDoS2019 e avaliam os algoritmos DT, *XGBoost* e RF. Na etapa de pré-processamento, os autores atribuem o valor da mediana a amostras com valores negativos e NaN, usam a escala de dados *StandardScaler* e fazem codificação com *label encoder* para cada classe começando do valor 0. Para a seleção de atributos, é usado o classificador *Extra Tree*, que

se baseia no Índice de Gini. São classificados 11 ataques presentes no CIC-DDoS2019, com exceção apenas do tipo *Portscan*. Os autores afirmam que o algoritmo *XGBoost* tem a maior acurácia, 98,72%, com seleção de atributos, e o RF obteve maior F1-*score* nos testes para detecção de tráfego benigno com 95% e uma pontuação maior que 95% para todos os outros ataques.

O presente artigo detecta e classifica ataques de DDoS usando o conjunto de dados CIC-DDoS2019, porém emprega uma metodologia diferente dos trabalhos anteriores. Os modelos de aprendizado de máquina desenvolvidos distinguem tráfego benigno de ataques DDoS, baseando-se apenas nos atributos dos pacotes trafegados. A abordagem exclui atributos de *socket* que possam facilitar a detecção, como a direção do fluxo (*Inbound*) e as portas de serviço, por exemplo, devido à facilidade com que administradores de rede podem alterá-las. É essencial remover esses atributos para evitar vieses nos modelos, garantir que eles aprendam padrões de comportamento genuínos de ataques DDoS e que eles obtenham capacidades de generalização para novos dados. Para contornar o desequilíbrio nas classes de tráfego benigno e de ataque no *dataset*, são propostos dois cenários de teste na classificação binária: um com subamostragem da classe de ataque e sobreamostragem da classe legítima via geração de dados sintéticos usando a técnica SMOTE, e outro apenas com subamostragem da classe de ataque. O estudo também realiza detecção multiclasse para sete dos treze tipos de ataques disponíveis, categorizados em UDP, SYN, NTP e OUTROS, que inclui os ataques LDAP, MSSQL, NETBIOS e DNS. O desempenho dos modelos NB, MLP, DT, RF e SVM são analisados em termos de tempo de execução e eficácia das métricas nas classificações binária e multiclasse.

3. Metodologia Proposta

As etapas da metodologia proposta para detecção e classificação de ataques DDoS estão indicadas na Figura 1. São cinco etapas: (i) Preparação dos Dados, (ii) Pré-processamento, (iii) Treinamento e Validação, (iv) Testes e (v) Avaliação do Modelo.

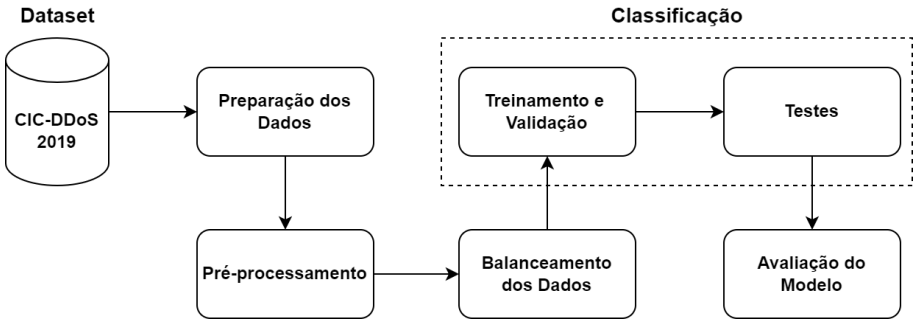


Figura 1. As etapas da metodologia proposta.

A Preparação dos Dados consiste na coleta dos dados do *dataset* CIC-DDoS2019, leitura e organização dos arquivos. Essa etapa é onde ocorre a primeira análise do conjunto de dados. Na etapa de Pré-processamento é feito o tratamento de dados faltantes (NaN) e valores infinitos, limpeza dos dados, aplicação de técnicas como *variance threshold*, *label encoder*, normalização e padronização de dados. A etapa de Treinamento e Validação é a primeira etapa da classificação dos dados. Nessa etapa, o *dataset* pré-processado é dividido em conjunto de treino e teste e é realizado o treinamento dos modelos de aprendizado de máquina escolhidos junto da validação dos dados. A etapa de

Testes é a segunda etapa da classificação de dados e consiste no teste efetivo dos dados não vistos durante o treinamento e validação. É a etapa na qual os modelos escolhidos tentam prever os dados classificando-os nas categorias possíveis. A Avaliação é a etapa final da metodologia proposta e nela se realiza a análise das predições do modelo na etapa de teste. Essa análise é feita através de métricas de desempenho como acurácia, precisão, *recall*, *f1-score* e ROC-AUC. A técnica de validação cruzada também é usada para avaliar o desempenho de alguns algoritmos. As seções seguintes detalham o conjunto de dados escolhido, as técnicas de pré-processamento, de seleção de atributos e de balanceamento dos dados e também apresentam os algoritmos de aprendizado de máquina usados na avaliação com os seus hiperparâmetros.

3.1. Conjunto de Dados CIC-DDoS2019

O conjunto de dados (*dataset*) usado na avaliação é o CIC-DDoS2019, disponibilizado pelo *Canadian Institute for Cybersecurity* (CIC) da *University of New Brunswick* (UNB), Canadá [Sharafaldin et al., 2019]. O *dataset* CIC-DDoS2019 consiste em 431.371 amostras, sendo 333.540 amostras rotuladas como "Ataque" e 97.831 rotuladas como "Benigna". Diferente de *datasets* anteriores, o CIC-DDoS2019 define uma nova taxonomia de ataques DDoS e corrige as deficiências atuais para ataques DDoS que podem ser realizados usando protocolos da camada de aplicação que usam TCP e/ou UDP.

Este trabalho utiliza o *dataset* no formato *Comma-Separated Values* (CSV). Ele contêm 19 arquivos divididos em dois diretórios, um para o Dia 1 (teste) e outro para o Dia 2 (treinamento). O Dia 1 contém 7 arquivos de ataque, sendo: PortMap, Net-BIOS, LDAP, MSSQL, UDP, UDP-Lag e SYN. Já o Dia 2 contém 12 arquivos de ataque: NTP, DNS, LDAP, MSSQL, NetBIOS, SNMP, SSDP, UDP, UDP-Lag, WebDDoS, SYN, TFTP. Os dados de ataques foram capturados durante um intervalo de 7 a 13 minutos, com exceção do UDP e TFTP que duraram 24 minutos e 3 horas e 40 minutos, respectivamente.

Os 13 ataques do CIC-DDoS2019 são divididos em categorias. Nos experimentos da classificação binária são considerados todos os tipos de ataques, exceto o WebDDoS porque o volume de tráfego contido no *dataset* é muito baixo, segundo os próprios autores do CIC-DDoS2019. Nos experimentos da classificação multiclasse, são considerados 7 ataques divididos em 4 categorias: UDP, SYN, NTP e OUTROS, a qual inclui os ataques LDAP, MSSQL, NETBIOS e DNS.

3.2. Pré-processamento de dados

O primeiro passo nesta etapa é remover linhas com valores infinitos, NaN, duplicadas e aquelas com valor 0 na coluna '*Protocol*'. São mantidas apenas as linhas com valores 6 (TCP) e 17 (UDP) nessa coluna, com enfoque em ataques na camada de transporte. A coluna '*Label*' é codificada como 0 (tráfego benigno) e 1 (tráfego maligno) para a classificação binária e com valores inteiros para a classificação multiclasse, exceto o algoritmo MLP, para o qual foi usada a codificação *one-hot-encode* para se adequar à arquitetura da rede neural. Os dados são normalizados no intervalo [-1, 1] usando *MinMaxScaler* do *sklearn* porque o trabalho implementa algoritmos sensíveis a escala como MLP e SVM, além do conjunto de dados conter características com diferentes unidades de medida, por exemplo, pacotes por segundo, bytes por segundo, comprimento e tamanho de pacote, etc. e alguns atributos conterem valores *outliers*. A normalização reduz

o impacto desses valores durante o treinamento dos modelos e garante que os atributos tenham um peso igual independente da unidade de medida.

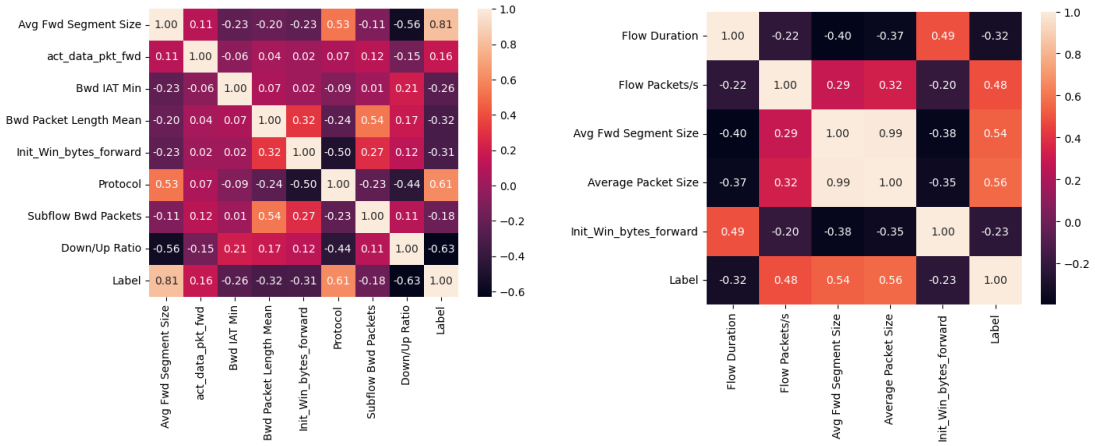
3.2.1. Seleção de Atributos

De um total de 89 atributos, primeiramente, são removidos os atributos *'Unnamed: 0.1', 'Unnamed: 0', 'Source IP', 'Destination IP', 'Source Port', 'Destination Port', 'Timestamp', 'SimillarHTTP', 'Flow ID', 'Inbound', 'Fwd Header Length.1'*, sobrando 78 atributos. Depois utiliza-se as técnicas *variance threshold* e *feature_importances_* do algoritmo RF implementado na biblioteca *sklearn* do *Python*. Com o *variance threshold*, para a classificação multiclasse, remove-se atributos que tenham uma variabilidade inferior a 20%, resultando em 58 colunas, sem contar com a coluna '*Label*' que é o alvo. Em seguida, através do RF, seleciona-se 30% do total de atributos mais importantes restando apenas 17 atributos. Para a classificação binária, aplica-se 0% de *variance threshold* e também seleciona-se 30% do total de atributos mais importantes com o *feature_importances_* do RF, restando somente 20 atributos.

Os valores de *variance threshold* e *feature_importances_* via RF foram escolhidos após diversos testes realizados variando a cada 10% os valores. A faixa de *variance threshold* testada foi de 0% até 30% enquanto a faixa do percentual da lista total de atributos importantes selecionados via RF foi de 10% até 40%. Foram anotados todos os resultados das combinações e escolhidos os valores que combinados resultavam nas melhores métricas. Então, no total foram realizados 16 testes para a classificação binária e 16 testes para a classificação multiclasse, em todos os modelos, nesta primeira etapa de seleção de atributos. A seleção de atributos via *Random Forest* é eficaz, pois o modelo cria diversas árvores de decisão em subconjuntos de treino e estima a importância de cada atributo pela diminuição da impureza de Gini ou ganho de informação. Após treinar todas as árvores, as pontuações dos atributos são agregadas e ordenadas, permitindo a redução da dimensionalidade dos dados e a execução mais rápida do treinamento e teste dos modelos.

Para a classificação binária, são escolhidos 20 atributos entre 66 disponíveis e gerada uma matriz de correlação para destacar as relações lineares entre eles. Observa-se alta correlação positiva, maior que 0,7, em alguns atributos, levando à aplicação de um filtro que seleciona apenas atributos com correlação inferior a 0,7. Isso reduz os atributos de 20 para 8, excluindo o atributo *'Label'*. Os atributos finais selecionados e a nova matriz de correlação são apresentados na Figura 2(a). Para a classificação multiclasse, são escolhidos 17 atributos de 58 restantes. Depois, aplicou-se o mesmo processo de filtro explicado na classificação binária, o que resultou em apenas 5 atributos na classificação multiclasse, conforme a Figura 2(b).

Ao remover atributos com alta correlação, todos os modelos são novamente testados para garantir que os resultados não sejam afetados. Observa-se que os atributos *'Average Packet Size'* e *'Avg Fwd Segment Size'* têm alta correlação positiva de 0,99, e remover um deles diminuía as métricas dos modelos. Isso pode ser devido a ataques específicos que aumentam o tamanho do pacote sem aumentar o tamanho do segmento, já que é possível injetar dados no campo "opções" do pacote dificultando a detecção por mecanismos de segurança. Portanto, decidiu-se manter os dois atributos, pois eles têm correlação média satisfatória com o alvo, 0,56 e 0,54, respectivamente, e removê-los re-



(a) Classificação binária. (b) Classificação multiclasse.

Figura 2. Matrizes de correlação dos atributos finais selecionados.

sultaria na perda de informações relevantes para a predição dos modelos.

3.3. Balanceamento do Conjunto de Dados

As técnicas de balanceamento do conjunto de dados também são diferentes para a classificação binária e para a classificação multiclasse. Para a classificação binária, são escolhidos 12 dos 13 ataques, excluindo o *WebDDoS* devido à baixa quantidade de amostras desse ataque presentes no *dataset* [Sharafaldin et al., 2019]. Além disso, o *dataset* possui uma quantidade muito menor de amostras de tráfego benigno do que de amostras de tráfego de ataques DDoS. Para balancear as classes, são consideradas duas abordagens. Na primeira, é feita uma sobreamostragem da classe minoritária usando a técnica SMOTE [Chawla et al., 2002], que gera dados sintéticos, e subamostragem da classe majoritária. Na segunda, é feita somente uma subamostragem da classe majoritária. No entanto, Chawla *et al.* afirmam que a combinação de SMOTE e subamostragem, tem melhor desempenho do que a subamostragem simples [Chawla et al., 2002]. Por isso, são definidos dois cenários para a classificação binária:

- **Cenário 1:** subamostragem da classe DDoS + sobreamostragem da classe de tráfego benigno com SMOTE.
- **Cenário 2:** subamostragem da classe DDoS.

Após a geração de dados sintéticos no Cenário 1, aplica-se um filtro de remoção de duplicatas em cada subconjunto de dados para garantir a ausência de dados redundantes possivelmente gerados pelo SMOTE. Esse processo de geração de dados sintéticos e remoção de duplicatas é repetido até que todas as amostras sejam diferentes. A Tabela 1 apresenta o resultado do balanceamento de carga para os Cenários 1 e 2.

Para a classificação multiclasse, são realizados testes com diferentes tipos de ataques DDoS e seleciona-se 7 dos 13 ataques. São escolhidos ataques das subcategorias TCP, TCP/UDP e UDP, abrangendo as categorias de reflexão e exploração. Após o balanceamento do *dataset*, os ataques LDAP, MSSQL, NETBIOS e DNS são agrupados na categoria “OUTROS”, enquanto UDP, SYN e NTP permanecem em categorias individuais, como apresentado na Tabela 2.

Tabela 1. Amostras de tráfego de ataque e benigno em cada cenário antes e depois da reamostragem de dados.

		Amostras		
Reamostragem	**Cenário**	**Ataque**	**Benigno**	**Proporção**
Antes	1 e 2	639.175	2.870	0,45%
Depois	1	320.000 30.000	320.000 30.000	
	2	2.870	2.870	

Tabela 2. Ataques escolhidos na classificação multiclasse e (re)amostras por classe.

Categoria	Baseado em	Tipo do Ataque	Amostras	Classe	(Re)amostras
Ataque de Reflexão	TCP	MSSQL	61.926	OUTROS	55.676
	TCP/UDP	DNS LDAP NETBIOS	15.552 21.704 13.919		
	UDP	NTP	56.767	NTP	13.919
Ataque de Exploração	TCP	SYN Flood	38.721	SYN	13.919
	UDP	UDP Flood	155.022	UDP	13.919

Para obter pelo menos 10 mil amostras por ataque, foi necessário ler os ataques LDAP e NETBIOS diretamente dos arquivos originais do CIC-DDoS2019, pois no subconjunto *"cicddos2019_5percent"* utilizado através da plataforma *Kaggle* [Kaggle, 2024], esses ataques têm menos de 10 mil amostras. Em seguida, os ataques são balanceados aplicando subamostragem simples com a classe *RandomUnderSampler* do *imblearn*, que seleciona aleatoriamente amostras das classes majoritárias até atingir a quantidade da menor classe (NETBIOS com 13.919 amostras). Por fim, o conjunto balanceado foi dividido 66% para treinamento e o restante para teste.

3.4. Modelos e Hiperparâmetros

Para o treinamento, são considerados 5 algoritmos de aprendizado de máquina supervisionado: NB, MLP, SVM, DT e RF. Todos os modelos são implementados na linguagem *Python*, versão 3.10.12, pela biblioteca *sklearn* [Pedregosa et al., 2011], exceto o MLP que é implementado pela biblioteca *keras* do *tensorflow*. Não foi utilizada nenhuma técnica de otimização de hiperparâmetros como *Grid Search* [Liashchynskyi e Liashchynskyi, 2019] ou *Nature Inspired Search* [Fister Jr et al., 2013]. Os parâmetros para cada modelo são escolhidos empiricamente após diversos testes manuais para aumentar o desempenho da classificação. A Tabela 3 resume todos os parâmetros e valores usados na classificação binária e multiclasse. Quando os parâmetros são diferentes para cada classificação o nome da classificação aparece entre parêntesis. O valor do parâmetro de regularização (C) do SVM pode ser visto direto nas Tabelas 3, 4, 5, 6 e 7. Também é usada a técnica de validação cruzada por *k-folds* para avaliar o desempenho dos algoritmos DT, RF e MLP, em ambas as classificações, com e sem seleção de atributos.

4. Resultados

Os resultados da avaliação dos algoritmos NB, MLP, SVM, DT e RF estão divididos em classificação binária e classificação multiclasse. Na classificação binária, são

Tabela 3. Parâmetros dos algoritmos para a classificação binária e multiclasse.

Algoritmo	Parâmetros
MLP	Solver: adam (Estimativa de momento adaptativo) Taxa de Aprendizado: 0,01 Neurônio por Camada (binária): 20, 10, 2 Neurônio por Camada (multiclasse): 32, 16, 8, 4 Função de Ativação (binária): relu, relu, softmax Função de Ativação (multiclasse): relu, relu, relu, softmax Percentual de Validação: 33% Máximo de Falhas de Validação: 6 Máximo de Iterações: 100 Embaralhar Dados (*shuffle*): True
SVM	Parâmetro de regularização (C): 75 Kernel: poly Coeficiente do Kernel: auto Tolerância para critério de parada: 0,001 Forma da função de decisão: one-vs-rest
Árvore de Decisão	Critério: gini Divisor: *best* Mínimo de amostras para divisão: 2 Mínimo de amostras para nó folha: 1
Random Forest	Critério: gini Número de árvores: 100 Mínimo de amostras para divisão: 2 Mínimo de amostras para nó folha: 1 Número de atributos considerados para divisão: *sqrt*

consideradas ainda os Cenários 1 e 2, descritos na Seção 3.3. As simulações foram realizadas na plataforma *Kaggle* [Kaggle, 2024].

4.1. Classificação Binária

4.1.1. Cenário 1: subamostragem da classe DDoS + sobreamostragem da classe de tráfego benigno com SMOTE

No Cenário 1, os algoritmos são executados considerando dois conjuntos de dados criados na etapa de balanceamento de dados, sendo um com 30.000 e outro com 320.000 amostras por classe (DDoS e benigno). A Tabela 4 mostra o resultado da avaliação dos algoritmos nesse cenário, sem o uso da técnica de seleção de atributos. O melhor desempenho, nos dois conjuntos, é obtido pelo RF. No conjunto de 320.000 amostras, o RF obteve 99,99% em quase todas as métricas e, no conjunto de 30.000 amostras, obteve 99,99% de precisão. Os algoritmos DT, MLP e SVM possuem métricas acima de 98% para os dois conjuntos. O NB com PCA obteve métricas a partir de 70% para os dois conjuntos.

A Tabela 5 mostra o resultado da avaliação dos algoritmos também no Cenário 1, para os dois conjuntos balanceados, porém com o uso da técnica de seleção de atributos definida na metodologia proposta. O resultado mais interessante é que todos os algoritmos, agora, possuem métricas superiores a 93%. Esse desempenho se deve à seleção de

Tabela 4. Resultados da classificação binária sem seleção de atributos (Cenário 1).

Amostras por Classe	Algoritmo	Acurácia	Precisão	Recall	F1	ROC-AUC
30.000	*Naive* Bayes (NB)	98,59%	99,15%	98,04%	98,59%	98,60%
	Naive Bayes (PCA = 11)	83,22%	77,59%	93,79%	84,93%	83,14%
	MLP	99,68%	99,63%	99,73%	99,68%	99,68%
	Árvore de Decisão (DT)	99,89%	99,92%	99,86%	99,89%	99,89%
	Random Forest (RF)	99,94%	99,99%	99,88%	99,94%	99,94%
	SVM (C = 1)	99,50%	99,75%	99,25%	99,50%	99,50%
320.000	*Naive* Bayes (NB)	98,41%	98,99%	97,82%	98,40%	98,41%
	Naive Bayes (PCA = 1)	78,60%	70,63%	97,74%	82,00%	78,64%
	MLP	99,75%	99,95%	99,56%	99,75%	99,75%
	Árvore de Decisão (DT)	99,98%	99,98%	99,98%	99,98%	99,98%
	Random Forest (RF)	99,99%	99,99%	99,98%	99,99%	99,99%
	SVM (C=1)	99,71%	99,73%	99,69%	99,71%	99,71%

atributos realizada em duas etapas: a primeira combina técnicas de limite de variância com o método *feature_importances_* do *Random Forest*, enquanto a segunda etapa é feita a partir da análise da matriz de correlação dos atributos, na qual se manteve apenas os atributos com baixa ou média correlação entre si, porém média ou alta correlação com o alvo. Atributos com alta correlação entre si podem fornecer informações redundantes e manter ambos pode prejudicar o desempenho do modelo devido à multicolinearidade. Já atributos com alta correlação com o alvo, são mais prováveis de serem úteis para prever o resultado desejado. Com a metodologia proposta, consegue-se chegar a um conjunto específico de atributos que fornecem informações úteis sobre o tráfego de rede, ajudando os algoritmos a detectarem ataques DDoS. Essa abordagem, aliada ao pré-processamento dos dados, é também o que resulta no desempenho superior do RF frente a outros algoritmos.

Tabela 5. Resultados da classificação binária com seleção de atributos (Cenário 1).

Amostras por Classe	Algoritmo	Acurácia	Precisão	Recall	F1	ROC-AUC
30.000	*Naive* Bayes (NB)	98,56%	99,20%	97,92%	98,56%	98,56%
	Naive Bayes (PCA = 3)	96,13%	99,66%	92,63%	96,02%	96,16%
	MLP	98,69%	99,91%	97,48%	98,68%	98,70%
	Árvore de Decisão (DT)	99,91%	99,88%	99,94%	99,91%	99,91%
	Random Forest (RF)	99,90%	99,92%	99,87%	99,90%	99,90%
	SVM (C = 1e7)	99,46%	99,10%	99,83%	99,47%	99,46%
320.000	*Naive* Bayes (NB)	98,38%	99,00%	97,78%	98,39%	98,39%
	Naive Bayes (PCA = 3)	96,34%	99,59%	93,13%	96,25%	96,37%
	MLP	99,58%	99,48%	99,68%	99,58%	99,57%
	Árvore de Decisão (DT)	99,95%	99,95%	99,96%	99,95%	99,95%
	Random Forest (RF)	99,95%	99,95%	99,96%	99,95%	99,95%
	SVM (C=1000)	98,61%	99,61%	97,62%	98,61%	98,62%

Por ter apresentado o melhor desempenho para classificação binária no Cenário 1, o tempo de detecção de ataques DDoS pelo algoritmo RF com 8 atributos é medido. Para o conjunto com 30.000 amostras, o tempo é de 0,14 s e para o conjunto de 320.000 amostras é de 1,59 s. Além disso, os resultados corroboram parcialmente à hipótese de que a

combinação de SMOTE e subamostragem simples tem melhor desempenho que apenas a subamostragem simples [Chawla et al., 2002]. MLP, DT e RF alcançaram melhores métricas no Cenário 1 (SMOTE + subamostragem simples), ao contrário de NB e SVM que tiveram desempenho inferior. Portanto, ao utilizar SMOTE, modelos como MLP, DT ou RF são recomendados para se obter melhores resultados.

4.1.2. Cenário 2: subamostragem da classe DDoS

No Cenário 2, os algoritmos são executados considerando um subconjunto de dados criado na etapa de balanceamento de dados, com 2.870 amostras por classe (DDoS e benigno). A Tabela 6 mostra o resultado da avaliação dos algoritmos nesse cenário, sem o uso da técnica de seleção de atributos. O melhor desempenho, neste subconjunto, é obtido pelo RF e pelo SVM empatados com 99,80% de acurácia e com pontuações parecidas no restante das métricas. O algoritmo menos eficiente é o NB com PCA, assim como no Cenário 1.

Tabela 6. Resultados da classificação binária sem seleção de atributos (Cenário 2).

Algoritmo	Acurácia	Precisão	*Recall*	F1	ROC-AUC
Naive Bayes (NB)	99,08%	99,47%	98,64%	99,05%	99,07%
Naive Bayes (PCA = 14)	81,61%	74,41%	95,19%	83,52%	81,88%
MLP	99,59%	99,48%	99,69%	99,58%	99,59%
Árvore de Decisão (DT)	99,74%	99,69%	99,79%	99,74%	99,74%
Random Forest (RF)	99,80%	99,79%	99,79%	99,79%	99,79%
SVM (C=300)	99,80%	99,69%	99,90%	99,79%	99,80%

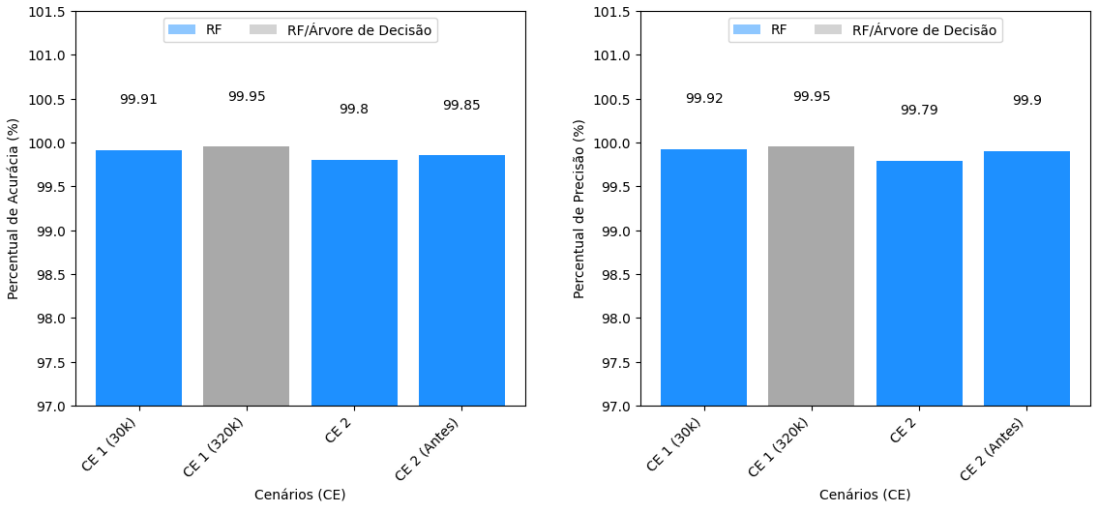
A Tabela 7 mostra o resultado da avaliação dos algoritmos também no Cenário 2, porém com o uso da técnica de seleção de atributos definida na metodologia proposta. Nota-se mais uma vez o efeito positivo da seleção de atributos em duas etapas considerada pela metodologia proposta. No Cenário 1, o desempenho do algoritmo NB com PCA aumentou significamente, pois com a metodologia proposta os atributos já foram previamente analisados e selecionados, logo, as informações mais importantes já estão concentradas nesses atributos. Além disso, também se observa uma ligeira diminuição nas métricas, que era esperado, quando comparado com os resultados sem seleção de atributos do Cenário 2, exceto com o algoritmo RF que não sofreu alterações nas pontuações, sendo o melhor algoritmo do Cenário 2 com o *dataset* reduzido para 2.870 amostras por classe.

Tabela 7. Resultados da classificação binária com seleção de atributos (Cenário 2).

Algoritmo	Acurácia	Precisão	*Recall*	F1	ROC-AUC
Naive Bayes (NB)	98,62%	98,95%	98,22%	98,58%	98,61%
Naive Bayes (PCA = 3)	97,08%	99,34%	94,67%	96,95%	97,03%
MLP	98,67%	99,26%	98,01%	98,63%	98,65%
Árvore de Decisão (DT)	99,69%	99,58%	99,79%	99,69%	99,69%
Random Forest (RF)	99,80%	99,79%	99,79%	99,79%	99,79%
SVM (C=1e7)	99,28%	98,76%	99,79%	99,27%	99,29%

Outra vez, por ter apresentado o melhor desempenho para classificação binária no Cenário 2, o tempo de detecção de ataques DDoS pelo algoritmo RF com 8 atributos é medido e o resultado é de 0,02 s.

A Figuras 3(a) e 3(b) apresentam a acurácia e a precisão, respectivamente, do melhor modelo em cada conjunto de dados nos Cenários 1 e 2, além do resultado do RF no Cenário 2 antes da segunda etapa de seleção de atributos. A cada etapa de seleção de atributos realizada, os testes são refeitos e os resultados anotados. Embora não caibam neste artigo devido a limitação de páginas, os resultados antes da segunda etapa de seleção de atributos demonstram que o RF, com 20 atributos, atinge 99,98% em quase todas as métricas com 320 mil amostras por classe e 99,90% com 30 mil amostras por classe, exceto a precisão que obteve 99,93%.



(a) Acurácia. (b) Precisão.

Figura 3. Métricas do melhor algoritmo nos Cenários 1 e 2.

4.2. Classificação Multiclasse

Na classificação multiclasse, considera-se as classes e o conjunto de dados definido na Tabela 2. A Tabela 8 mostra os resultados da avaliação dos cinco algoritmos, sem a técnica de seleção de atributos. Observa-se que os algoritmos baseados em árvore de decisão possuem o melhor desempenho, com resultados acima de 99%, destacando-se o RF. Em seguida, o MLP supera ligeiramente o SVM, enquanto o NB é menos eficiente, com resultados entre 80% e 90%, exceto pelo *recall* do NB com PCA, que atingiu 93,05%.

A Tabela 9 mostra resultados para a classificação multiclasse com a seleção de atributos definida pela metodologia proposta. É feita ainda uma distinção entre os resultados obtidos antes e depois da segunda etapa da seleção de atributos. As linhas cinza nessa tabela representam os resultados considerando 17 atributos e as linhas brancas representam os resultados considerando 5 atributos. A seleção de atributos antes da análise da matriz de correlação melhora os resultados gerais dos algoritmos em comparação com a ausência de seleção (Tabela 8), especialmente para o algorimo NB. Após a análise da matriz de correlação, há uma leve queda nos resultados dos algoritmos, exceto para o

Tabela 8. Resultados da classificação multiclasse sem seleção de atributos.

Algoritmo	Acurácia	Precisão	*Recall*	F1
Naive Bayes (NB)	82,29%	81,76%	88,73%	82,68%
Naive Bayes (PCA = 10)	89,17%	86,98%	93,05%	89,10%
MLP	98,37%	97,97%	98,15%	98,04%
Árvore de Decisão (DT)	99,18%	99,07%	99,12%	99,10%
Random Forest (RF)	99,36%	99,40%	99,19%	99,30%
SVM (C=75)	98,26%	97,76%	98,13%	97,92%

NB sem PCA, que tem uma queda maior. Isso indica que a redução de atributos empregada tornou os algoritmos mais eficientes, já que os modelos aprendem mais rápido com um conjunto menor de atributos, sem comprometer significativamente o desempenho dos resultados, destacando-se novamente o RF, cujas métricas ultrapassam 99%.

Tabela 9. Resultados da classificação multiclasse com seleção de atributos.

Algoritmo	Matriz Correlação	Acurácia	Precisão	*Recall*	F1
Naive Bayes (NB)	Antes	93,35%	90,88%	95,91%	92,67%
	Depois	79,19%	84,87%	71,72%	62,76%
Naive Bayes (PCA = 10)	Antes	92,90%	90,32%	95,22%	92,34%
Naive Bayes (PCA = 2)	Depois	93,70%	90,67%	95,00%	92,65%
MLP	Antes	98,74%	98,64%	98,52%	98,57%
	Depois	97,90%	97,31%	97,33%	97,30%
Árvore de Decisão (DT)	Antes	99,21%	99,14%	99,12%	99,13%
	Depois	99,10%	99,02%	98,97%	99,00%
Random Forest (RF)	Antes	99,34%	99,38%	99,16%	99,27%
	Depois	99,21%	99,23%	99,01%	99,12%
SVM (C=100)	Antes	98,47%	98,16%	98,22%	98,16%
SVM (C=1000)	Depois	97,97%	97,39%	97,40%	97,39%

4.3. Discussão

A seleção de atributos da metodologia proposta resultou em um conjunto reduzido e específico de *features*, tornando-a mais eficiente em comparação com outras metodologias que também usaram o conjunto de dados CIC-DDoS2019. Esse é o caso dos trabalhos de Elsayed *et al.* [Elsayed et al., 2020] e de Nazarudeen e Sundar [Nazarudeen e Sundar, 2022]. É importante ressaltar que não são realizados experimentos comparativos entre as metodologias porque os dois trabalhos citados não disponibilizam os *scripts* e conjuntos de dados usados em suas avaliações. Os conjuntos de dados e *scripts* usados na avaliação deste artigo estão disponíveis, respectivamente, nas plataformas *Kaggle* [1] [2] e *Github* [3].

Na classificação binária, Elsayed *et al.* introduzem o sistema DDoSNet, que combina Redes Neurais Recorrentes (RNN) com *autoencoder*, alcançando 99% de acurácia e 98,8% de ROC-AUC em um conjunto de dados com um total de 230.673 amostras, sendo

[1] https://www.kaggle.com/datasets/manmandes/cicddos2019-5percent
[2] https://www.kaggle.com/datasets/rodrigorosasilva/cic-ddos2019-30gb-full-dataset-csv-files
[3] https://github.com/rsrodrigo/SBSeg2024-deteccao-de-ataque-ddos-cic-ddos2019

23.000 amostras de teste. Com a metodologia proposta, o algoritmo RF no Cenário 1, alcança 99,9% de acurácia e ROC-AUC para um conjunto de dados com 30.000 amostras por classe, e 99,95% de acurácia e ROC-AUC para o conjunto de dados com 320.000 amostras por classe. Para o Cenário 2 com seleção de atributos, o RF com a metodologia proposta e considerando 8 atributos tem acurácia de 99,8%. Esse valor de acurácia é superior ao valor alcançado pelos algoritmos avaliados por Elsayed *et al.* [Elsayed et al., 2020] e por Nazarudeen e Sundar [Nazarudeen e Sundar, 2022].

A precisão do melhor modelo em cada cenário deste artigo, comparada com os melhores modelos dos trabalhos relacionados, mostra que para o Cenário 1 com os dois conjuntos de dados (30 mil e 320 mil amostras), a metodologia proposta obteve os melhores resultados. Sobre a precisão no Cenário 2, antes da segunda etapa de seleção de atributos, o RF apresenta precisão superior ao trabalho de Elsayed *et al.* [Elsayed et al., 2020].

Nazarudeen e Sundar [Nazarudeen e Sundar, 2022] realizam a classificação multiclasse com o RF, alcançando entre 0,94 e 1 em todas as métricas para todos os ataques e 100% para o ataque SYN [Nazarudeen e Sundar, 2022]. A Figura 4 mostra que o algoritmo RF com metodologia proposta, após a análise da matriz de correlação, obteve resultados equivalentes, com 100% de detecção para o ataque SYN e pontuações de 0,98 ou acima para UDP, NTP e OUTROS (LDAP, MSSQL, NETBIOS e DNS). Em termos de tempo de detecção, o RF com 5 atributos detecta 33.128 amostras em 0,35 s.

	precision	recall	f1-score	support
NTP	0.99	0.99	0.99	4772
OUTROS	0.99	0.99	0.99	18919
SYN	1.00	1.00	1.00	4753
UDP	0.99	0.98	0.99	4684
accuracy			0.99	33128
macro avg	0.99	0.99	0.99	33128
weighted avg	0.99	0.99	0.99	33128

Figura 4. Relatório de classificação do RF com a metodologia proposta, após análise da matriz de correlação.

5. Conclusão e Trabalhos Futuros

Este artigo propôs uma metodologia para detectar e classificar ataques DDoS baseada em algoritmos de aprendizado de máquina. A metodologia proposta emprega técnicas de balanceamento de dados, pré-processamento e seleção de atributos. Os diferenciais da metodologia proposta incluem a possibilidade de balanceamento do conjunto de dados com a técnica SMOTE, uma etapa de pré-processamento de dados, na qual não são considerados atributos *socket* específicos de rede e que indicam a entrada ou saída do fluxo de rede, e a seleção de atributos realizada em duas etapas. Com a metodologia proposta, foram avaliados os algoritmos *Naive Bayes* (NB), *MultiLayer Perceptron* (MLP), Árvore de Decisão (*Decision Tree* - DT), *Random Forest* (RF) e *Support Vector Machine* (SVM) para o conjunto de dados CIC-DDoS2019.

Entre os algoritmos avaliados, o *Random Forest* (RF) se destacou, apresentando o melhor desempenho tanto na classificação binária (Cenários 1 e 2) quanto na classificação multiclasse. Na classificação binária, o RF tem um acurácia superior a 99,8%.

Na classificação multiclasse, o RF alcança métricas superiores a 99% no geral. Os resultados também corroboraram parcialmente a hipótese de que a combinação de SMOTE e subamostragem simples tem melhor desempenho que apenas a subamostragem simples [Chawla et al., 2002]. Foram identificados também indícios de que a metodologia proposta é mais eficiente do que metodologias propostas na literatura, que também usaram o conjunto de dados CIC-DDoS2019. O RF com a metodologia proposta e considerando 8 atributos tem acurácia de 99,8%, que é superior ao valor alcançado pelos algoritmos avaliados por Elsayed *et al.* [Elsayed et al., 2020] e por Nazarudeen e Sundar [Nazarudeen e Sundar, 2022].

Os trabalhos futuros incluem a implementação de outros algoritmos de aprendizado de máquina, como KNN, a validação do modelo em sistemas reais para comparação de desempenho e testes em ambientes reais com outras ferramentas de inteligência artificial para mitigação de ataques DDoS. Além disso, busca-se aprimorar as técnicas de pré-processamento de dados para obter melhores resultados nas métricas e explorar outros conjuntos de dados, como o CIC-IDS2017, que contenham diferentes tipos de ataques para aumentar a cobertura e eficácia na detecção de ataques DDoS.

Agradecimentos

Este trabalho foi realizado com recursos da RNP, CNPq, CEFET/RJ, CAPES, FAPERJ e PGC/UFF.

Referências

Agiollo, A., Bardhi, E., Conti, M., Lazzeretti, R., Losiouk, E. e Omicini, A. (2023). GNN4IFA: Interest flooding attack detection with graph neural networks. Em *IEEE European Symposium on Security and Privacy (EuroS&P)*, p. 615–630.

Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L. e Rieck, K. (2022). Dos and don'ts of machine learning in computer security. Em *USENIX Security Symposium*.

Bala, B. e Behal, S. (2024). AI techniques for IoT-based DDoS attack detection: Taxonomies, comprehensive review and research challenges. *Computer science review*, 52:100631.

Chawla, N. V., Bowyer, K. W., Hall, L. O. e Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research*, 16:321–357.

Elsayed, M. S., Le-Khac, N.-A., Dev, S. e Jurcut, A. D. (2020). DDoSNet: A deep-learning model for detecting network attacks. Em *IEEE 21st International Symposium on "A World of Wireless, Mobile and Multimedia Networks" (WoWMoM)*, p. 391–396.

Fister Jr, I., Yang, X.-S., Fister, I., Brest, J. e Fister, D. (2013). A brief review of nature-inspired algorithms for optimization. *arXiv preprint arXiv:1307.4186*.

Horchulhack, P., Viegas, E., Santin, A. e Geremias, J. (2022). Atualização de modelo baseado em aumento de dados e transferência de aprendizagem para detecção de intrusão em redes. Em *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, p. 223–235.

Kaggle (2024). Level up with the largest AI & ML community. https://www.kaggle.com/. (Acesso em 19 de junho de 2024).

Kurniabudi, Stiawan, D., Darmawijoyo, Idris, M. Y. B., Bamhdi, A. M. e Budiarto, R. (2020). CICIDS-2017 dataset feature analysis with information gain for anomaly detection. *IEEE Access*, 8:132911–132921.

Laufer, R. P., Moraes, I. M., Velloso, P. B., Bicudo, M. D. D., Campista, M. E. M., Cunha, D. O., Costa, L. H. M. K. e Duarte, O. C. M. B. (2005). Negação de serviço: Ataques e contramedidas. Em *Minicursos do Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, p. 1–63.

Li, Z., Kong, Y. e Jiang, C. (2023). A transfer double deep Q network based DDoS detection method for internet of vehicles. *IEEE Transactions on Vehicular Technology*, 72(4):5317–5331.

Liashchynskyi, P. e Liashchynskyi, P. (2019). Grid search, random search, genetic algorithm: a big comparison for nas. *arXiv preprint arXiv:1912.06059*.

Lima, M., Neira, A., Borges, L. e Nogueira, M. (2023). Predição não-supervisionada de ataques DDoS por sinais precoces e one-class SVM. Em *Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais (SBSeg)*, p. 403–416.

Nazarudeen, F. e Sundar, S. (2022). Efficient DDoS attack detection using machine learning techniques. Em *IEEE International Power and Renewable Energy Conference (IPRECON)*, p. 1–6.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., , Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M. e Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *The Journal of machine Learning research*, 12:2825–2830.

Polat, H., Polat, O. e Cetin, A. (2020). Detecting DDoS attacks in software-defined networks through feature selection methods and machine learning models. *Sustainability*, 12(3):1035.

Sharafaldin, I., Lashkari, A. H., Hakak, S., e Ghorbani, A. A. (2019). Developing realistic distributed denial of service (DDoS) attack dataset and taxonomy. Em *IEEE 53rd International Carnahan Conference on Security Technology*.

Yoachimik, O., Desgats, J. e Forster, A. (2023). Cloudflare mitigates record-breaking 71 million request-per-second DDoS attack. https://blog.cloudflare.com/cloudflare-mitigates-record-breaking-71-million-request-per-second-ddos-attack/. (Acesso em 25 de agosto 2023).