v7.n6.2026

REVISTA CIENTÍFICA - RECIMA21 ISSN 2675-6218

AVALIAÇÃO DE ALGORITMOS DE APRENDIZADO DE MÁQUINA PARA DETECÇÃO DE ATAQUES DDOS EM AMBIENTES *WORDPRESS* CONTEINERIZADOS

EVALUATION OF MACHINE LEARNING ALGORITHMS FOR DDOS ATTACK DETECTION IN CONTAINERIZED WORDPRESS ENVIRONMENTS

EVALUACIÓN DE ALGORITMOS DE APRENDIZAJE AUTOMÁTICO PARA LA DETECCIÓN DE ATAQUES DDOS EN ENTORNOS WORDPRESS CONTENERIZADOS

Nadianne Maria dos Santos Galvão[1], Andre Luis Meneses Silva[2]

e768286
https://doi.org/10.47820/recima21.v7i6.8286

PUBLICADO: 06/2026

RESUMO

Este artigo avalia a aplicação de algoritmos de *machine learning* na detecção de ataques distribuídos de negação de serviço (*DDoS*) em ambiente *WordPress* conteinerizado. O estudo parte do problema da disponibilidade em aplicações *web* baseadas em sistemas de gerenciamento de conteúdo, especialmente quando a *API REST* e serviços como *WooCommerce* são expostos a acessos massivos e automatizados. Foram avaliados os algoritmos *Random Forest*, Regressão Logística e *Support Vector Machine* (*SVM*), utilizando bases públicas do *CIC-DDoS2019* e dados reais gerados em ambiente controlado. O ambiente experimental foi composto por *WordPress* e banco de dados em *Docker*, simulações de carga com *Locust*, captura de pacotes com *tcpdump* e extração de fluxos com *CICFlowMeter*. Os resultados indicaram desempenho superior do *Random Forest*, com acurácia e *F1-score* iguais a 1,00 nos principais cenários balanceados e no conjunto real de ataque. Regressão Logística e *SVM* apresentaram desempenho inferior nos dados reais balanceados, com *F1-score* médio próximo de 0,48. Conclui-se que modelos baseados em árvores são promissores para detecção precoce de *DDoS* em *APIs WordPress* conteinerizadas, desde que integrados a monitoramento contínuo e políticas de resposta.

PALAVRAS-CHAVE: Aprendizado de Máquina. DDoS. *WordPress*. *Docker*. *Random Forest.*

ABSTRACT

This article evaluates the use of machine learning algorithms for detecting distributed denial-of-service (DDoS) attacks in a containerized WordPress environment. The study addresses the availability problem in web applications based on content management systems, especially when REST APIs and services such as WooCommerce are exposed to massive and automated access. Random Forest, Logistic Regression, and Support Vector Machine (SVM) were evaluated using public CIC-DDoS2019 datasets and real traffic data generated in a controlled environment. The experimental environment included WordPress and database containers, workload simulation with Locust, packet capture with tcpdump, and network-flow extraction using CICFlowMeter. The results showed superior performance for Random Forest, reaching accuracy and F1-score of 1.00 in the main balanced scenarios and in the real attack dataset. Logistic Regression and SVM achieved lower performance on the balanced real dataset, with average F1-score close to 0.48. The findings indicate that tree-based models are promising for early DDoS detection in

[1] Universidade Federal de Sergipe, Mestranda em Ciência da Computação.
[2] Universidade Federal de Sergipe, Douto em Engenharia Elétrica

containerized WordPress APIs when combined with continuous monitoring and automated response mechanisms.

KEYWORDS: *Machine Learning. DDoS. WordPress. Docker. Random Forest.*

RESUMEN
Este artículo evalúa la aplicación de algoritmos de aprendizaje automático en la detección de ataques distribuidos de denegación de servicio (DDoS) en un entorno WordPress contenerizado. El estudio aborda el problema de la disponibilidad en aplicaciones web basadas en sistemas de gestión de contenido, especialmente cuando la API REST y servicios como WooCommerce están expuestos a accesos masivos y automatizados. Se evaluaron los algoritmos Random Forest, Regresión Logística y Support Vector Machine (SVM), utilizando bases públicas CIC-DDoS2019 y datos reales generados en un entorno controlado. El ambiente experimental incluyó WordPress y base de datos en Docker, simulaciones de carga con Locust, captura de paquetes con tcpdump y extracción de flujos con CICFlowMeter. Los resultados indicaron un rendimiento superior de Random Forest, con exactitud y F1-score de 1,00 en los principales escenarios balanceados y en el conjunto real de ataque. Regresión Logística y SVM obtuvieron resultados inferiores en los datos reales balanceados, con F1-score promedio cercano a 0,48. Se concluye que los modelos basados en árboles son prometedores para la detección temprana de DDoS en APIs WordPress contenerizadas.

PALABRAS CLAVE*: Aprendizaje Automático. DDoS. WordPress. Docker. Random Forest.*

INTRODUÇÃO

De acordo com Castells (2020), a *internet* teve origem em um contexto marcado por interesses militares em paralelo a iniciativas científicas, impulsionada pela Agência de Projetos de Pesquisa Avançada (ARPA), na década de 1960, com a proposta de criar uma rede descentralizada capaz de transmitir mensagens sem depender de centros fixos de controle. Com o tempo, essa rede passou a ser utilizada também no meio acadêmico e recebeu o nome de *Internet*. Com seu crescimento, foi necessário criar um padrão de comunicação que permitisse a troca de mensagens através de dispositivos diferentes, que ficou conhecido como protocolo *TCP*.

Assim, torna-se mais fácil compreender o conceito de *internet*, que, segundo Tanenbaum (2021), corresponde a um vasto conjunto de redes heterogêneas interconectadas, as quais utilizam protocolos padronizados para fornecer, de forma integrada, uma variedade de serviços específicos.

A partir dos anos de 1990, a criação da *World Wide Web* por Tim Berners-Lee e o surgimento dos primeiros *sites* e *blogs* ampliaram o uso da *internet* como espaço de informação, seja ela pessoal ou corporativa. Com a popularização dos *websites*, surgiram os Sistemas de Gerenciamento de Conteúdo, ou *Content Management Systems* (*CMS*), voltados à criação, edição e organização de *sites* de forma mais acessível (DialHost, 2018). Entre os principais *CMS* disponíveis, destacam-se *WordPress*, *Joomla* (Joomla, 2025) e *Drupal* (Drupal, 2025).

O *WordPress* destaca-se pela ampla adoção, flexibilidade de temas e *plugins* e integração com serviços externos por meio de *APIs* (Hostnet, 2024). Entre seus recursos, a *API REST* permite a interação com conteúdos e dados da aplicação por meio de requisições *HTTP* e manipulação de dados em *JSON*. Porém, os *endpoints REST* podem ser submetidos a grandes volumes de requisições, o que torna a disponibilidade um aspecto crítico. Quando esse volume é intencionalmente gerado com o objetivo de degradar ou tornar indisponível um serviço, tem-se um ataque de negação de serviço distribuído, ou *DDoS*.

De acordo com a Amazon Web Services (AWS, 2024), os ataques distribuídos de negação de serviço, do inglês *Distributed Denial of Service*, ou simplesmente *DDoS*, consistem em sobrecarregar um sistema com um grande volume de pacotes ou dados. Esse tipo de ataque compromete um dos princípios fundamentais da cibersegurança: o princípio da disponibilidade. Esse tipo de ataque busca consumir recursos computacionais, como *CPU*, memória, conexões de rede, banco de dados e filas de processamento. Em aplicações *web*, ataques direcionados à camada de aplicação podem ser especialmente danosos, pois requisições aparentemente legítimas podem acionar rotinas custosas, consultas ao banco de dados ou o processamento de regras de negócio. Assim, um ambiente pode permanecer “no ar” do ponto de vista do servidor, mas tornar-se praticamente inacessível ao usuário final devido à alta latência e ao aumento de erros *HTTP*.

Uma forma de aumentar o isolamento em serviços de hospedagem é utilizar virtualização. Máquinas virtuais abstraem recursos de *hardware* em ambientes de execução independentes (Silberschatz; Galvin; Gagne, 2018). Já os contêineres representam uma alternativa mais leve, pois operam no nível do sistema operacional, favorecendo portabilidade, atualização, recuperação e escalabilidade (Microsoft Azure, 2025; Amazon Web Services, 2025). Mas a conteinerização não elimina o risco de indisponibilidade. Um ambiente *WordPress* executado em contêiner pode continuar vulnerável a picos intensos de tráfego, principalmente quando a aplicação depende de banco de dados, *plugins* e *endpoints* dinâmicos. Dessa forma, torna-se relevante investigar como mecanismos de detecção podem auxiliar na identificação antecipada de padrões anômalos.

No contexto da cibersegurança, o aprendizado de máquina tem se destacado como ferramenta para antecipar cenários de risco e identificar padrões associados a ameaças (Cisco, 2025). Como campo da Inteligência Artificial, essa abordagem utiliza dados e algoritmos para reconhecer padrões e apoiar decisões em cenários complexos (Google Cloud, 2025).

Diante desse contexto, este artigo tem como problema de pesquisa a seguinte questão: qual é a eficácia de algoritmos de aprendizado de máquina na detecção de ataques *DDoS* em

ambientes *WordPress* conteinerizados, com foco em tráfego direcionado à *API REST* do *WooCommerce*? A questão se embasa no uso expressivo de *WordPress* em ambientes reais, na crescente adoção de contêineres para implantação de aplicações e na necessidade de mecanismos capazes de diferenciar tráfego legítimo de tráfego malicioso em situações de sobrecarga.

O objetivo geral deste estudo é comparar a eficiência de algoritmos de aprendizado de máquina na detecção de ataques *DDoS* em um ambiente *WordPress* conteinerizado, com foco no tráfego direcionado à *API REST* do *WooCommerce*. Como objetivos específicos, busca-se: I) Identificar vulnerabilidades do *WordPress* em cenários de ataques *DDoS*, especialmente em *endpoints* da *API REST*; II) Realizar simulações controladas de tráfego legítimo e malicioso; III) Treinar e validar modelos de aprendizado de máquina, como *Random Forest*, Regressão Logística e *Support Vector Machine* (*SVM*), para a detecção de tráfego anômalo; IV) Avaliar o desempenho dos modelos por meio de métricas como acurácia, *F1-score* e tempo de resposta; V) Analisar o impacto dos ataques no ambiente conteinerizado, considerando aspectos como latência, escalabilidade e consumo de recursos computacionais.

Assim, o artigo contribui ao apresentar uma avaliação prática e experimental sobre o uso de aprendizado de máquina em um cenário aplicado de segurança cibernética, combinando ambiente conteinerizado, *CMS WordPress*, *API REST*, simulação de tráfego e classificação de ataques. Dessa forma, os experimentos aproximam-se de uma realidade comum a muitas organizações que utilizam *WordPress* como base de suas aplicações *web* e buscam mecanismos de detecção de ataques voltados à indisponibilidade de serviços.

1. REFERENCIAL TEÓRICO

Este tópico apresenta os principais conceitos relacionados ao trabalho, como *WordPress*, *API REST* e *WooCommerce*, contêineres e *Docker*, ataques *DDoS* e aprendizado de máquina aplicado à detecção de tráfego anômalo. Esses conceitos são importantes para compreender o ambiente utilizado na pesquisa e os procedimentos aplicados na detecção de ataques *DDoS* em ambiente *WordPress* conteinerizado. Em seguida tem-se uma apresentação dos estudos correlacionados.

1.1. Categorias Conceituais

A primeira categoria conceitual corresponde a *WordPress*, *API REST* e *WooCommerce*. De acordo com McKeown (2015), o *WordPress* surgiu a partir do código-fonte do *b2*, com o objetivo inicial de facilitar a publicação de *blogs*. Com o tempo, a plataforma evoluiu para um

sistema de gerenciamento de conteúdo amplamente utilizado, incorporando temas, *plugins* e recursos de personalização. Segundo a Hostnet (2024), o *WordPress* destaca-se pela ampla adoção e pela possibilidade de integração com serviços externos.

No contexto do *WordPress*, a *API REST* permite que aplicações interajam com conteúdos e informações da plataforma por meio de requisições *HTTP* e dados em formato *JSON*. Essa funcionalidade amplia as possibilidades de integração, mas também expõe *endpoints* que podem ser alvo de acessos automatizados e tráfego excessivo, especialmente em aplicações com *WooCommerce*.

Na sequência tem-se os *Contêineres e Docker* como segunda categoria conceitual. De acordo com Vitalino e Castro (2018), contêineres agrupam uma aplicação e suas dependências, compartilhando o *kernel* do sistema operacional hospedeiro. Siddiqui, Siddiqui e Khan (2019) também caracterizam os contêineres como mecanismos de empacotamento de aplicações, voltados à portabilidade e à eficiência operacional.

Diferentemente das máquinas virtuais, que abstraem recursos de *hardware* e executam sistemas operacionais completos em ambientes isolados, os contêineres operam de forma mais leve, compartilhando o *kernel* do sistema hospedeiro. Garrison e Nova (2017) destacam que a virtualização tradicional, baseada em *hypervisors*, permite executar múltiplas máquinas virtuais em um mesmo servidor físico, mas com maior consumo de recursos quando comparada à abordagem baseada em contêineres.

Nesse contexto, o *Docker* destaca-se como uma das principais plataformas para criação, gerenciamento e execução de contêineres. Segundo a *Red Hat* (2023), a plataforma utiliza recursos do *kernel Linux* para isolar processos e permitir a execução independente de múltiplos serviços. Vitalino e Castro (2018) acrescentam que as imagens *Docker* são compostas por camadas, e os contêineres representam instâncias em execução dessas imagens.

Como terceira categoria temos os Ataques Distribuídos de Negação de Serviço que representam uma ameaça direta à disponibilidade, um dos pilares da segurança da informação. Segundo Galegale (2017), a disponibilidade busca garantir que as informações estejam acessíveis a usuários autorizados sempre que necessário.

Gomes, Araujo e Campos (2015) destacam que ataques de negação de serviço passaram a ser registrados com maior relevância a partir da década de 1990, afetando provedores, universidades e grandes portais. Esses incidentes demonstraram que ataques *DDoS* podem comprometer diretamente a continuidade de serviços digitais.

De acordo com a *IBM* (2025), esses ataques podem ser classificados conforme o tipo de recurso explorado e a técnica utilizada. Entre eles, destacam-se os ataques na camada de

aplicação, que buscam comprometer aplicações *web* por meio do envio excessivo de requisições maliciosas; os ataques de protocolo, direcionados às camadas 3 e 4 do Modelo *OSI*, afetando recursos fundamentais da comunicação em rede; os ataques volumétricos, caracterizados pela inundação da rede com alto volume de tráfego; e os ataques multivetoriais, que combinam múltiplas abordagens com o objetivo de potencializar os danos e dificultar a mitigação.

Por fim, temos o aprendizado de máquina na detecção de ataques. Segundo a *IBM* (2024), o aprendizado de máquina é um ramo da Inteligência Artificial e da Ciência da Computação que utiliza dados e algoritmos para permitir que sistemas aprendam padrões e apoiem decisões. Essa abordagem pode ser classificada em aprendizado supervisionado, quando utiliza dados rotulados; não supervisionado, quando busca padrões em dados não rotulados; e semi-supervisionado, quando combina dados rotulados e não rotulados.

Brown (2021) destaca que o aprendizado de máquina tende a impactar diferentes setores, exigindo compreensão de seus potenciais e limitações. Na cibersegurança, essa abordagem pode auxiliar na identificação de padrões associados a ameaças e tráfegos anômalos.

Kuncheva (2004) aponta elementos fundamentais do aprendizado supervisionado, como classes, características ou atributos e base de dados. Esses elementos definem o objetivo da classificação, descrevem os dados analisados e sustentam o treinamento e a avaliação dos modelos.

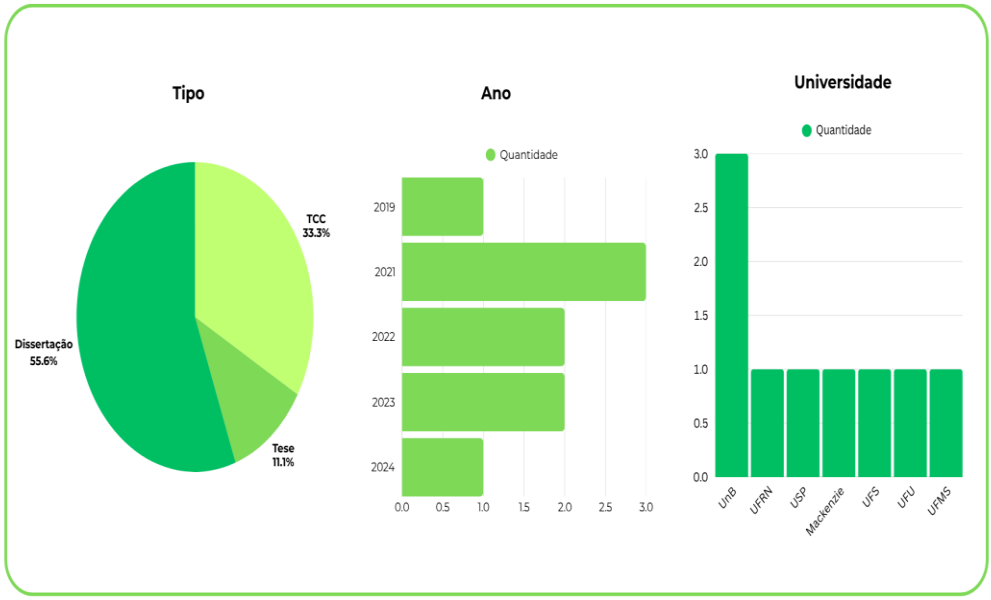
Henke *et al*. (2021) observam que o crescimento dos ataques evidencia limitações em soluções tradicionais de mitigação, motivando o uso de aprendizado de máquina para identificação de padrões anômalos. Neste trabalho, são avaliados *Random Forest*, Regressão Logística e *Support Vector Machine*. O *Random Forest* combina múltiplas árvores de decisão; a Regressão Logística é aplicada principalmente a problemas de classificação binária; e o *SVM* busca separar classes por meio de um hiperplano com margem máxima.

1.2. Trabalhos relacionados

Foram selecionados nove trabalhos relacionados aos objetivos desta pesquisa, os quais abordam, principalmente, o uso de aprendizado de máquina, inteligência artificial e técnicas de análise de tráfego para detecção ou mitigação de ataques *DDoS*. A análise desses estudos mostrou a recorrência das palavras-chave “aprendizagem de máquina” e “*DDoS*”, evidenciando o alinhamento com a temática deste artigo.

Para sintetizar o perfil dos estudos selecionados, a Figura 1 apresenta a distribuição dos trabalhos relacionados quanto ao tipo de produção acadêmica, ao ano de publicação e à universidade de origem.

Figura 1. Caracterização dos trabalhos relacionados



Fonte: Elaboração própria (2026).

Entre os trabalhos analisados, Almeida Neto (2021) propôs uma abordagem baseada na combinação de aprendizado de máquina e entropia para detecção de ataques *DDoS* em redes definidas por *software*, obtendo taxa de detecção superior a 99% e falso positivo inferior a 1%. Machado (2023) discutiu o papel da inteligência artificial no aperfeiçoamento e na mitigação de ataques em redes *SDN*, concluindo que não há uma solução única, mas sim a necessidade de combinar diferentes estratégias conforme o contexto da organização.

Teles (2022) comparou métodos para detecção de ataques *DDoS* utilizando a base *UNB CIC-IDS 2017*, combinando *PCA* com *ANN* e *SVM*, alcançando *F-scores* próximos a 100%. Ferreira (2021), por sua vez, avaliou algoritmos como *KNN*, *SVM*, Florestas Randômicas, Árvores de Decisão e Redes Neurais, obtendo melhor desempenho com Florestas Randômicas e Árvores de Decisão. Já Santos Neto (2021) investigou algoritmos não supervisionados para detecção de *DDoS* em ambientes de redes definidas por *software*.

Com o intuito de avaliar os aumentos de ataques cibernéticos, especialmente os *DDoS*, Figueiredo *et al.* (2022) verificaram a eficácia de diferentes técnicas de aprendizado de máquina na detecção de ataques *DDoS*, com o objetivo de desenvolver um sistema capaz de prevenir e mitigar esses ataques. Concluíram que o algoritmo de árvore de decisão é uma ferramenta promissora para a detecção de ataques *DDoS*.

No que se refere à relevância da seleção de variáveis na otimização de modelos voltados à detecção de ataques *DDoS*, Araújo (2023) observou que o algoritmo *XGBoost* demonstrou elevada robustez, enquanto a técnica *ANOVA* destacou-se como um método eficaz de *feature selection*. Além disso, embora a classificação multiclasse apresente maior complexidade e demande mais tempo de treinamento em comparação à classificação binária, mostrou-se vantajosa por fornecer informações mais detalhadas sobre os diferentes tipos de ataques.

Lima Filho (2019) desenvolveu um estudo aprofundado sobre a concepção e implementação de um sistema de segurança de redes voltado à detecção e mitigação de ataques *DDoS*. O sistema, denominado *Smart Defender*, integra as ferramentas *Smart Detection* e *Smart Protection*, e demonstrou-se eficaz na identificação e contenção desse tipo de ameaça, promovendo maior resiliência em redes e serviços *online*. A arquitetura distribuída do sistema, aliada ao emprego de técnicas de aprendizado de máquina, evidencia seu potencial como ferramenta estratégica no enfrentamento do crescente número de ataques de negação de serviço.

Chagas (2024) direcionou sua investigação ao estudo dos ataques de negação de serviço (*DoS*) em bancos de dados, destacando-os como uma ameaça relevante à disponibilidade dos sistemas de informação. A detecção desses ataques apresenta desafios consideráveis, sobretudo devido à complexidade dos *logs* gerados pelos sistemas gerenciadores de banco de dados e à escassez de ferramentas especializadas para essa finalidade. O autor propôs uma solução baseada na aplicação de técnicas de aprendizado de máquina na análise de *logs* de consultas *SQL*, demonstrando a eficácia dessa abordagem na identificação de ataques *DoS* em ambientes de banco de dados.

Os trabalhos apresentados mostram o papel central do aprendizado de máquina para detectar e mitigar ataques, com diferentes algoritmos que apresentaram bom desempenho em diferentes estudos. Vale salientar que foi reforçado que não há uma abordagem universal para a mitigação de ataques *DDoS*, mas a necessidade de avaliar cada caso e assim criar combinações específicas de técnicas ajustadas ao contexto, seja ele de rede ou sistema. Adicionalmente, os trabalhos apresentados mostram um caminho sobre treinamento e agregam positivamente à

abordagem deste estudo. Contudo, o diferencial desta pesquisa para as anteriores é que serão aplicadas técnicas de aprendizado de máquina no contexto de contêineres e *WordPress*.

Diferentemente de parte dos trabalhos relacionados, que concentram suas análises em redes definidas por *software*, *datasets* públicos ou simulações genéricas de tráfego, este estudo direciona a investigação para um ambiente *WordPress* conteinerizado, com foco na *API REST* do *WooCommerce*. Dessa forma, busca-se aproximar a detecção de ataques *DDoS* de um cenário comum em aplicações *web* reais, especialmente em *sites* institucionais e lojas virtuais baseadas em *CMS*. Para isso, são utilizados os algoritmos *Random Forest*, Regressão Logística e *Support Vector Machine* (*SVM*), com o objetivo de avaliar seu desempenho na identificação de padrões associados a esse tipo de ameaça.

2. METODOLOGIA

Esta pesquisa possui caráter exploratório e abordagem quantitativa. É exploratória porque investiga a aplicação de algoritmos de aprendizado de máquina em um cenário específico: a detecção de ataques DDoS em ambiente *WordPress* conteinerizado, com foco na API REST do *WooCommerce.* É quantitativa porque compara o desempenho dos algoritmos por meio de métricas mensuráveis, como acurácia, precisão, recall e F1-score.

O estudo foi desenvolvido em duas etapas complementares. A primeira consistiu no treinamento e avaliação dos modelos utilizando *datasets* públicos do CIC-DDoS2019. Foram selecionados ataques do tipo SYN Flood, DrDoS UDP, UDP-Lag, DrDoS DNS e DrDoS NTP, com o objetivo de analisar o comportamento dos algoritmos diante de diferentes padrões de ataques DDoS. O Quadro 1 sintetiza os ataques considerados nessa etapa, destacando suas principais características e o objetivo de avaliação associado a cada um deles.

Quadro 1. Ataques e bases consideradas na etapa de treinamento

Ataque	Característica principal	Objetivo da avaliação
SYN Flood	Explora o *handshake* TCP por múltiplos pacotes SYN não finalizados.	Avaliar detecção de ataque clássico de exaustão de conexões.
DrDoS UDP	Usa reflexão/amplificação com servidores UDP intermediários.	Analisar tráfego UDP amplificado e massivo.
UDP-Lag	Gera latência por envio intenso de pacotes UDP.	Observar identificação de padrões ligados a atraso e instabilidade.
DrDoS DNS	Explora servidores DNS para amplificar tráfego contra a vítima.	Testar detecção em *dataset* altamente desbalanceado.

Ataque	Característica principal	Objetivo da avaliação
DrDoS NTP	Explora servidores NTP públicos mal configurados.	Avaliar separação entre tráfego benigno e malicioso em ataque de amplificação.

Fonte: Elaboração própria (2026).

Os dados passaram por etapas de pré-processamento, incluindo a remoção de colunas irrelevantes ou identificadoras, como endereços IP, portas, identificadores de fluxo e *timestamp*. A variável *Label* foi utilizada como alvo da classificação, assumindo valor 0 para tráfego benigno e 1 para tráfego de ataque. A Figura 2 apresenta o fluxo geral da metodologia adotada, desde a utilização dos *datasets* públicos e configuração do ambiente experimental até a avaliação dos modelos de aprendizado de máquina.

Figura 2. Fluxo metodológico da pesquisa para geração, captura, tratamento e classificação do tráfego em ambiente *WordPress*/WooCommerce conteinerizado

Fonte: Elaboração própria (2026).

De forma complementar, o Quadro 2 sintetiza as etapas executadas, indicando a função de cada fase no processo de treinamento, simulação, captura, extração e avaliação dos dados.

Quadro 2. Etapas do fluxo metodológico

Etapa	Descrição
1. Preparação dos dados públicos	Seleção dos ataques, limpeza de colunas e definição de *Label* como variável alvo.
2. Treinamento dos modelos	Aplicação de *Random Forest*, Regressão Logística e SVM com e sem balanceamento.
3. Simulação em *WordPress*	Execução de *WordPress*/*WooCommerce* em *Docker* e geração de carga com Locust.
4. Captura e extração	Coleta com tcpdump, inspeção com *Wireshark* e extração de fluxos com *CICFlowMeter.*
5. Avaliação comparativa	Comparação por acurácia, precisão, recall, F1-score, macro média e média ponderada.

Fonte: Elaboração própria (2026).

A segunda etapa consistiu na criação de um ambiente experimental controlado, composto por *WordPress* e *WooCommerce* executados em contêineres *Docker*, juntamente com o serviço de banco de dados e volumes persistentes. Esse ambiente foi utilizado para simular acessos massivos à API REST do *WooCommerce*, aproximando o experimento de um cenário real de sobrecarga em aplicações web baseadas em CMS. Para a geração de carga, foi utilizada a ferramenta Locust, responsável por simular múltiplos usuários realizando requisições simultâneas a *endpoints* da aplicação, como recursos relacionados a pedidos, clientes e produtos.

Durante os testes, o tráfego de rede foi capturado com o tcpdump, armazenado em arquivos no formato .pcap e posteriormente inspecionado com o *Wireshark.* Além disso, métricas dos contêineres foram acompanhadas por meio do *docker stats*, permitindo observar o consumo de CPU, memória e rede durante as simulações. Em seguida, os arquivos .pcap foram processados com o *CICFlowMeter,* ferramenta utilizada para extrair características dos fluxos de rede e convertê-las em arquivos .csv.

Após a extração dos dados reais, foi realizada uma análise de compatibilidade entre esses dados e os *datasets* públicos previamente tratados. A partir dessa comparação, definiu-se um conjunto de colunas em comum, permitindo que os modelos treinados fossem aplicados aos dados reais sem a necessidade de reestruturação completa do pipeline. Entre as características consideradas estavam duração do fluxo, intervalo entre pacotes, tamanho médio dos pacotes, quantidade de *bytes*, contagem de pacotes e métricas relacionadas aos subfluxos.

Os modelos foram implementados em Python, com apoio da biblioteca *scikit-learn.* Foram avaliados três algoritmos de aprendizado de máquina: *Random Forest*, Regressão Logística e *Support Vector Machine* (SVM). Para cada algoritmo, foram realizados experimentos em cenários com e sem balanceamento de classes. Nos cenários sem balanceamento, manteve-se a distribuição original dos dados, aproximando-se de situações reais, porém com possibilidade de viés para a classe majoritária. Já nos cenários com balanceamento, foi aplicada a técnica de *oversampling* por duplicação de amostras da classe minoritária, buscando equilibrar a quantidade de registros benignos e maliciosos.

A avaliação dos modelos foi realizada com base nas métricas de acurácia, precisão, recall e F1-score. A acurácia mede a proporção geral de classificações corretas, sendo definida por:

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

A precisão indica a proporção de classificações positivas que foram corretamente identificadas:

$$Precisão = \frac{VP}{VP + FP}$$

O *recall* mede a capacidade do modelo de identificar corretamente os ataques existentes:

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

Por fim, o F1-*score* representa o equilíbrio entre precisão e *recall:*

$$F1 = 2\,x\,\frac{(Precisão \times Recall)}{(Precisão + Recall)}$$

Nessas fórmulas, VP representa os verdadeiros positivos, VN os verdadeiros negativos, FP os falsos positivos e FN os falsos negativos. Além dessas métricas, também foram analisadas a macro média e a média ponderada, utilizadas para interpretar melhor o desempenho dos modelos em cenários com distribuição desigual entre as classes.

No experimento com dados reais, foi gerado um cenário de ataque com usuários simultâneos direcionados à API REST do *WooCommerce.* Esse cenário permitiu avaliar o comportamento dos modelos diante de tráfego produzido em ambiente controlado, considerando características próximas às de uma aplicação *WordPress* em funcionamento. Dessa forma, a metodologia possibilitou comparar o desempenho dos algoritmos tanto em bases públicas quanto em dados reais, permitindo uma avaliação mais ampla da detecção de ataques DDoS em ambientes *WordPress* conteinerizados.

3. RESULTADOS E DISCUSSÃO

Os resultados demonstraram que o desempenho dos modelos variou conforme o tipo de ataque, o nível de balanceamento das classes e a capacidade de cada algoritmo em diferenciar tráfego benigno de tráfego malicioso. Observou-se que a acurácia, quando analisada isoladamente, nem sempre representava adequadamente o desempenho dos modelos, especialmente em bases desbalanceadas, nas quais a classe majoritária podia influenciar a taxa geral de acertos.

Por esse motivo, a análise considerou também precisão, *recall*, *F1-score*, *macro média* e média ponderada. Essas métricas permitiram avaliar com maior clareza o comportamento dos algoritmos diante de classes desiguais, principalmente nos casos em que havia dificuldade de identificação da classe benigna.

Nos testes com os *datasets* públicos do *CIC-DDoS2019*, o *Random Forest* apresentou o melhor desempenho geral. Nos ataques *SYN Flood*, *DrDoS UDP*, *UDP-Lag*, *DrDoS DNS* e *DrDoS NTP*, o modelo manteve resultados elevados, especialmente após o balanceamento dos dados. Em diversos cenários, alcançou valores próximos ou iguais a 1,00 para acurácia, precisão, *recall* e *F1-score*, indicando boa capacidade de identificação dos padrões associados aos ataques avaliados.

A Regressão Logística apresentou bons resultados em alguns cenários, principalmente após o balanceamento, mas demonstrou maior sensibilidade à desproporção entre as classes. Em certos conjuntos, a acurácia geral foi elevada, enquanto o desempenho para a classe benigna foi inferior, evidenciando dificuldade de generalização para a classe minoritária.

O *SVM* apresentou comportamento semelhante. Em bases desbalanceadas, os resultados pareciam satisfatórios quando observada apenas a acurácia; porém, a análise do *F1-score* mostrou limitações na identificação da classe benigna. Após o balanceamento, houve melhora em parte dos experimentos, embora o modelo tenha permanecido sensível à distribuição dos dados e ao tipo de ataque analisado. A Tabela 1 sintetiza os principais resultados obtidos com os *datasets* públicos, destacando o comportamento dos três algoritmos avaliados em cada tipo de ataque.

Tabela 1. Síntese dos resultados obtidos com *datasets* públicos

Cenário	*Random Forest*	Regressão Logística	SVM	Observação
SYN Flood	F1 macro até 1,00 com balanceamento	Até 1,00 em Syn 03.11 balanceado	De 0,50 sem balanceamento a até 1,00 com balanceamento	Acurácia isolada ocultou falhas na classe benigna.
DrDoS DNS	F1 macro 1,00 com e sem balanceamento	F1 macro até 1,00/0,99 com balanceamento	F1 macro 0,76 sem e 0,99 com balanceamento	Balanceamento melhorou fortemente SVM.
DrDoS NTP	F1 macro 1,00 com e sem balanceamento	F1 macro 0,85 sem e 0,92 com balanceamento	F1 macro 0,55 sem e 0,95 com balanceamento	*Random Forest* foi mais estável.
UDP/UDP-Lag	Alto desempenho, com melhora na classe benigna após balanceamento	Melhora após balanceamento	Sensível à distribuição das classes	*Oversampling* reduziu viés para a classe majoritária.

Fonte: Elaboração própria (2026).

Conforme apresentado na Tabela 1, o *Random Forest* foi o modelo mais estável entre os algoritmos avaliados, mantendo desempenho elevado nos diferentes cenários. A Regressão Logística e o *SVM* também apresentaram bons resultados em alguns casos, mas dependeram mais do balanceamento e da separação entre as classes.

Além dos testes com *datasets* públicos, foi realizada uma avaliação com dados reais gerados em ambiente *WordPress* com *WooCommerce* executado em contêiner *Docker*. Para isso, o *Locust* foi utilizado para gerar acessos massivos à *API REST*, enquanto o tráfego foi capturado, armazenado e convertido em fluxos de rede por meio do *CICFlowMeter*. Essa etapa buscou aproximar o experimento de um cenário prático de sobrecarga em aplicação *web*. A Tabela 2 apresenta a síntese dos resultados obtidos com os dados reais gerados no ambiente experimental.

Tabela 2. Resultados com dados reais gerados no ambiente *WordPress*/WooCommerce

Modelo	Base avaliada	Acurácia	F1-score macro/médio	Interpretação
Random Forest	real_ddos_10000.csv com 6.516 fluxos de ataque	1,00	1,00	Reconheceu integralmente os fluxos maliciosos capturados.
Regressão Logística	*Dataset* real balanceado artificialmente	0,49	0,48	Dificuldade de separação entre classes simuladas.
SVM	*Dataset* real balanceado artificialmente	0,48	0,48	Desempenho próximo ao aleatório no cenário balanceado real.

No conjunto *real_ddos_10000.csv*, gerado a partir da simulação com usuários simultâneos, o *Random Forest* manteve o melhor desempenho, com precisão, *recall*, *F1-score* e acurácia geral iguais a 1,00 para a classe de ataque. Esse resultado indica que o algoritmo reconheceu com alta precisão os fluxos maliciosos gerados durante a simulação.

Entretanto, esse conjunto possuía apenas fluxos rotulados como ataque, o que limita a avaliação completa da capacidade do modelo de distinguir tráfego malicioso de tráfego benigno. Para permitir a execução de modelos que exigem duas classes, foi realizada uma estratégia de balanceamento artificial, duplicando amostras de ataque e atribuindo parte delas à classe

benigna simulada. Como essas amostras apresentavam características semelhantes, os modelos tiveram dificuldade em aprender diferenças consistentes entre as classes.

Nesse cenário, a Regressão Logística obteve acurácia de 0,49 e *F1-score* médio de 0,48, enquanto o *SVM* obteve acurácia de 0,48 e *F1-score* médio de 0,48. Esses resultados indicam desempenho próximo ao aleatório no conjunto real balanceado artificialmente, reforçando que a qualidade da separação entre as classes influencia diretamente o desempenho dos modelos.

O desempenho superior do *Random Forest* pode estar relacionado à sua capacidade de trabalhar com múltiplas árvores de decisão e capturar relações não lineares entre variáveis dos fluxos de rede, como duração do fluxo, quantidade de pacotes, tamanho médio dos pacotes, comportamento das *flags* e métricas de subfluxos.

Assim como foi observado no trabalho de Nicola, Lauretto e Delgado (2021), que avaliou classificadores e métodos de balanceamento na detecção de fraudes em transações com cartões de crédito, o modelo *Random Forest* também demonstrou superioridade neste estudo. Na pesquisa mencionada, os melhores resultados foram obtidos com *Random Forest* tanto em conjuntos desbalanceados quanto em cenários balanceados por sobreamostragem ou estratégias híbridas, sendo destacado ainda por sua robustez frente à escolha do método de balanceamento e à redução de atributos.

O que é válido salientar é que em cenários de tráfego *DDoS*, nos quais os padrões podem variar conforme o tipo de ataque, esse algoritmo demonstrou maior estabilidade.

De modo geral, os experimentos reforçam três pontos principais. Primeiro, o balanceamento dos dados impacta diretamente a qualidade da classificação. Segundo, a acurácia não deve ser analisada isoladamente, pois pode mascarar dificuldades na identificação da classe minoritária. Terceiro, o uso de dados reais é importante para aproximar a pesquisa de um cenário prático, mas esses dados devem conter tráfego benigno e malicioso para permitir uma avaliação mais completa.

Apesar dos bons resultados obtidos com o *Random Forest*, destaca-se como limitação que o conjunto real gerado representou principalmente o comportamento de ataque, sem quantidade equivalente de tráfego benigno real coletado no mesmo ambiente. Portanto, a avaliação com dados reais confirmou a capacidade do modelo de reconhecer padrões associados ao ataque simulado, mas ainda não permite afirmar definitivamente seu comportamento em um cenário completo, com usuários legítimos e atacantes atuando simultaneamente.

A principal contribuição deste estudo está na combinação entre bases públicas de ataques *DDoS* e um ambiente prático baseado em *WordPress*, *WooCommerce* e *Docker*.

Diferentemente de pesquisas concentradas em redes definidas por *software*, *datasets* públicos ou simulações genéricas, este trabalho direciona a análise para um cenário comum em aplicações *web* reais, como *sites* institucionais e lojas virtuais baseadas em *CMS*.

Em um cenário prático, os resultados indicam que modelos como o *Random Forest* poderiam ser utilizados como parte de uma estratégia de monitoramento, auxiliando na identificação de tráfego anômalo antes que a aplicação se torne indisponível. No entanto, essa aplicação exigiria integração com ferramentas de coleta contínua, definição de limiares de alerta e políticas de resposta, como bloqueio temporário de *IPs*, limitação de requisições ou acionamento de mecanismos de mitigação.

4. CONSIDERAÇÕES FINAIS

Os resultados obtidos indicam que o uso de algoritmos de aprendizado de máquina pode contribuir para a detecção de ataques *DDoS* direcionados a ambientes *WordPress* conteinerizados. Entre os modelos avaliados, *Random Forest*, Regressão Logística e *Support Vector Machine* (*SVM*), o *Random Forest* apresentou o melhor desempenho geral, demonstrando maior estabilidade nos diferentes cenários analisados, tanto nos *datasets* públicos quanto nos dados gerados no ambiente experimental.

Esse desempenho pode estar relacionado à capacidade do *Random Forest* de lidar com padrões complexos e não lineares, além de sua robustez diante de características comuns em fluxos de rede sob ataque, como variações na quantidade de pacotes, duração dos fluxos, tamanho médio dos pacotes e comportamento das *flags*. Em comparação, a Regressão Logística e o *SVM* apresentaram desempenho mais dependente do balanceamento dos dados e da separação entre as classes, mostrando maior sensibilidade em cenários com distribuição desigual ou com características semelhantes entre tráfego benigno e malicioso.

Além dos testes com *datasets* públicos, a realização de simulações com *Locust* permitiu observar o comportamento de um ambiente *WordPress* com WooCommerce executado em contêiner *Docker* diante de acessos massivos à *API REST*. Embora o *site* não tenha ficado completamente indisponível durante os testes mais intensos, foram observados aumento de latência e ocorrência de erros *HTTP 500*, indicando que ataques de negação de serviço podem comprometer a disponibilidade e afetar diretamente a experiência de acesso, mesmo em ambientes conteinerizados.

Como diferencial positivo, este trabalho combinou a utilização de bases públicas consolidadas, como o CIC-DDoS2019, com a geração de tráfego em um ambiente

WordPress/*WooCommerce* conteinerizado. Essa abordagem permitiu avaliar os modelos não apenas em *datasets* previamente tratados, mas também em um cenário controlado mais próximo de aplicações *web* reais, como *sites* institucionais e lojas virtuais baseadas em *CMS*. Dessa forma, a pesquisa contribui ao aproximar a detecção de ataques DDoS de um contexto prático, com foco na API REST do *WooCommerce.*

Apesar dos resultados promissores, é importante destacar uma limitação da etapa com dados reais. O conjunto gerado a partir da simulação representou principalmente o comportamento de ataque, não contendo uma quantidade equivalente de tráfego benigno real coletado no mesmo ambiente. Assim, embora os resultados indiquem que o modelo foi capaz de reconhecer padrões associados ao ataque simulado, ainda não é possível afirmar de forma definitiva seu desempenho em um cenário completo, com usuários legítimos e atacantes atuando simultaneamente.

Diante disso, conclui-se que modelos de aprendizado de máquina, especialmente algoritmos baseados em árvores, como o *Random Forest*, representam uma alternativa promissora para auxiliar na detecção precoce de ataques *DDoS* em *APIs* de aplicações *web*. Em um cenário prático, esse tipo de modelo poderia ser integrado a mecanismos de monitoramento contínuo, contribuindo para a identificação de tráfego anômalo antes que a aplicação se torne indisponível.

Como trabalhos futuros, recomenda-se ampliar a coleta de dados reais, contemplando tráfego benigno e malicioso gerado no mesmo ambiente, a fim de permitir uma avaliação mais completa da capacidade de classificação dos modelos. Também se sugere realizar testes comparativos entre ambientes conteinerizados e ambientes virtualizados com máquinas virtuais, analisando diferenças relacionadas ao consumo de recursos, resposta à sobrecarga e tolerância a ataques de negação de serviço. Além disso, a avaliação de algoritmos mais avançados, como *XGBoost* e *LightGBM*, pode contribuir para verificar possíveis ganhos de desempenho, velocidade de inferência e capacidade de generalização em cenários de detecção de *DDoS*.

5. DECLARAÇÃO DE USO DE INTELIGÊNCIA ARTIFICIAL GENERATIVA

Declaro, para os devidos fins, que utilizei Inteligência Artificial Generativa, especificamente o *ChatGPT* em sua versão gratuita, para dar suporte às seguintes etapas da pesquisa: auxílio no fichamento de artigos e revisão ortográfica do texto. Os autores assumem integral responsabilidade pelo conteúdo intelectual e técnico do trabalho, em conformidade com a Portaria CNPq n.º 2.664/2026.

REFERÊNCIAS

ALMEIDA NETO, J. R. **Detecção de ataques DDoS em ambientes SDN/NFV utilizando algoritmos de aprendizagem de máquina não supervisionados em fluxos de dados**. 2021. Dissertação (Mestrado) – Universidade Federal de Sergipe, São Cristóvão, 2021. Disponível em: https://ri.ufs.br/handle/riufs/15022. Acesso em: 25 mar. 2025.

AMAZON WEB SERVICES. **O que é um ataque DDoS?** 2024. Disponível em: https://aws.amazon.com/pt/shield/ddos-attack-protection. Acesso em: 18 dez. 2024.

AMAZON WEB SERVICES. **Qual a diferença entre contêineres e máquinas virtuais?** 2025. Disponível em: https://aws.amazon.com/pt/compare/the-difference-between-containers-and-virtual-machines/. Acesso em: 25 mar. 2025.

ARAÚJO, P. **Impacto de métodos de seleção de variáveis na classificação de ataques DDoS utilizando XGBoost**. 2023. Dissertação (Mestrado) – Universidade de São Paulo, São Paulo, 2023. Disponível em: https://www.teses.usp.br/teses/disponiveis/3/3142/tde-21092023-082915/pt-br.php. Acesso em: 26 mar. 2025.

BROWN, S. **Machine Learning, Explained**. MIT Sloan School of Management, 2021. Disponível em: https://mitsloan.mit.edu/ideas-made-to-matter/machine-learning-explained. Acesso em: 24 abr. 2025.

CASTELLS, M. **A sociedade em rede**. São Paulo: Paz e Terra, 2020.

CHAGAS, D. M. **Detecção de ataques de negação de serviço em SGBDs a partir de logs internos usando abordagens supervisionada e não supervisionada**. 2024. Dissertação (Mestrado) – Universidade de Brasília, Brasília, 2024. Disponível em: http://repositorio2.unb.br/jspui/bitstream/10482/48503/1/2024_DaniloAndersonDeMouraChagas_DISSERT.pdf. Acesso em: 25 mar. 2025.

CISCO. **What is machine learning in security?** 2025. Disponível em: https://www.cisco.com/c/en/us/products/security/machine-learning-security.html. Acesso em: 25 mar. 2025.

DIALHOST. **O que é CMS, como funcionam e quais são os mais utilizados.** 2018. Disponível em: https://www.dialhost.com.br/blog/o-que-e-cms/. Acesso em: 25 mar. 2025.

DRUPAL. **Drupal**. 2025. Disponível em: https://new.drupal.org/home. Acesso em: 17 fev. 2025.

FERREIRA, M. **Detecção de DDoS por aprendizado de máquina**. 2021. Monografia (Trabalho de Conclusão de Curso) – Universidade de Brasília, Brasília, 2021. Disponível em: https://bdm.unb.br/bitstream/10483/29831/1/2021_Matheus_Siade_Ferreira_tcc.pdf. Acesso em: 25 mar. 2025.

FIGUEIREDO, B. et al. **Estudo e investigação de técnicas de IA para detecção de ataques DDoS**. 2022. Monografia (Trabalho de Conclusão de Curso). Disponível em: https://adelpha-api.mackenzie.br/server/api/core/bitstreams/2d6937e5-5768-4f24-adcb-c7d728ecf48b/content. Acesso em: 26 mar. 2025.

GALEGALE, N. V. Uma contribuição para a segurança da informação: um estudo de casos múltiplos com organizações brasileiras. **Perspectivas em Ciência da Informação**, v. 22, n. 3, p. 75-97, jul./set. 2017. Disponível em: https://www.scielo.br/j/pci/a/Srp97XX3Hyb4MfjxRH9gDgd/?format=pdf&lang=pt.

GARRISON, J.; NOVA, K. **Cloud Native Infrastructure**. [S.l.]: O'Reilly Media, 2017.

GOMES, L. de C.; ARAUJO, M. S. de A.; CAMPOS, V. S. **Negação de serviço e botnets**. 2015. EEL878 – Redes de Computadores I. Professor: Otto Carlos Muniz Bandeira Duarte. Disponível em: https://www.gta.ufrj.br/grad/15_1/dos/pages/hist.html.

GOOGLE CLOUD. **Aprendizado supervisionado e não supervisionado: qual é a diferença?** 2025. Disponível em: https://cloud.google.com/discover/supervised-vs-unsupervised-learning. Acesso em: 25 mar. 2025.

HENKE, M. et al. Aprendizagem de máquina para segurança em redes de computadores: métodos e aplicações. In: **SIMPÓSIO BRASILEIRO EM SEGURANÇA DA INFORMAÇÃO E DE SISTEMAS COMPUTACIONAIS**, 11., 2021, Manaus. Anais [...]. Manaus: [s.n.], 2021. Disponível em: https://books-sol.sbc.org.br/index.php/sbc/catalog/download/95/419/690?inline=1. Acesso em: 25 mar. 2025.

HOSTNET. **Por que escolher o WordPress como a plataforma do seu site?** 2024. Disponível em: https://www.hostnet.com.br/blog/por-que-escolher-o-wordpress-como-a-plataforma-do-seu-site/. Acesso em: 25 mar. 2025.

IBM. **What is Machine Learning?** 2024. Disponível em: https://www.ibm.com/think/topics/machine-learning. Acesso em: 25 mar. 2025.

IBM. **O que é um ataque distributed denial-of-service (DDoS)?** 2025. Disponível em: https://www.ibm.com/br-pt/topics/ddos. Acesso em: 25 mar. 2025.

JOOMLA. **Joomla content management system (CMS)**. 2025. Disponível em: https://www.joomla.org/. Acesso em: 17 fev. 2025.

KUNCHEVA, L. **Combining Pattern Classifiers: methods and algorithms**. [S.l.]: Wiley-Interscience, 2004.

LIMA FILHO, F. **Smart Defender: um sistema de detecção e mitigação de ataques DoS/DDoS usando aprendizagem de máquina**. 2019. Tese (Doutorado) – Universidade Federal do Rio Grande do Norte, Natal, 2019. Disponível em: https://repositorio.ufrn.br/jspui/handle/123456789/28470. Acesso em: 25 mar. 2025.

MACHADO, L. R. **Inteligência artificial para identificar e tratar ataques de negação de serviço em redes baseadas em software**. 2023. Monografia (Trabalho de Conclusão de Curso). Disponível em: https://repositorio.ufms.br/handle/123456789/8147. Acesso em: 25 mar. 2025.

MCKEOWN, S. **Milestones: the story of WordPress**. WordPress.org, 2015. Disponível em: https://wordpress.org/book/.

MICROSOFT AZURE. **Máquinas virtuais: computadores virtuais dentro de computadores**. 2025. Disponível em: https://azure.microsoft.com/pt-br/resources/cloud-computing-dictionary/what-is-a-virtual-machine. Acesso em: 25 mar. 2025.

NICOLA, V. G. O. M.; LAURETTO, M. S.; DELGADO, K. V. **A**valiação empírica de classificadores e métodos de balanceamento para detecção de fraudes em transações com cartões de crédito. In: **ENCONTRO NACIONAL DE INTELIGÊNCIA ARTIFICIAL E COMPUTACIONAL, 17.**. Anais [...]. [S.l.: s.n.], 2021. p. 1-12. Disponível em: https://sol.sbc.org.br/index.php/eniac/article/view/12118/11983. Acesso em: 25 maio 2025.

RED HAT. **Docker: desenvolvimento de aplicações em containers**. 2023. Disponível em: https://www.redhat.com/pt-br/topics/containers/what-is-docker. Acesso em: 19 jan. 2023.

SANTOS NETO, M. J. **Detecção de ataque DDoS em SDN utilizando entropia e machine learning**. 2021. Dissertação (Mestrado) – Universidade de Brasília, Brasília, 2021. Disponível em: http://icts.unb.br/jspui/handle/10482/40991. Acesso em: 26 mar. 2025.

SIDDIQUI, T.; SIDDIQUI, S. A.; KHAN, N. A. Comprehensive analysis of container technology. In: **INTERNATIONAL CONFERENCE ON INFORMATION SYSTEMS AND COMPUTER NETWORKS (ISCON)**, 4., 2019, Mathura. Proceedings [...]. Mathura: GLA University, 2019. Disponível em: https://www.researchgate.net/profile/Tamanna-Siddiqui-2/publication/339976396_Comprehensive_Analysis_of_Container_Technology/links/61fd960c702c892cef04c5af/Comprehensive-Analysis-of-Container-Technology.pdf. Acesso em: 25 mar. 2025.

SILBERSCHATZ, A.; GALVIN, P. B.; GAGNE, G. **Operating System Concepts**. 10. ed. Hoboken: Wiley, 2018. ISBN 978-1-118-06333-0. Disponível em: http://os-book.com/OS10/index.html.

TANENBAUM, A. S. **Computer Networks**. [S.l.]: Pearson, 2021.

TELES, J. G. N. **Detecção de ameaças DDoS com aprendizagem de máquina**. 2022. Trabalho de Conclusão de Curso – Universidade Federal de Uberlândia, Uberlândia, 2022. Disponível em: https://repositorio.ufu.br/bitstream/123456789/34622/1/DeteccaoAmeacasDDoS.pdf. Acesso em: 25 mar. 2025.

UNIVERSITY OF NEW BRUNSWICK. **DDoS Evaluation Dataset (CIC-DDoS2019)**. Canadian Institute for Cybersecurity. Disponível em: https://www.unb.ca/cic/datasets/ddos-2019.html. Acesso em: 25 mar. 2025.

VITALINO, C.; CASTRO, P. **Descomplicando o Docker**. Brasil: Brasport, 2018.