Avaliação de Algoritmos de Aprendizado de Máquina para Detecção de Ataques DDoS em Ambientes WordPress Baseados em Contêineres

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

[1]Universidade Federal de Sergipe (UFS), Departamento de Sistema de Informação - Av. Ver. Olímpio Grande, s/n - Porto, Itabaiana - SE

[2]Universidade Federal de Sergipe (UFS), Departamento de Sistema de Informação - Av. Ver. Olímpio Grande, s/n - Porto, Itabaiana - SE

[nadiannegalvao@gmail.com, andrelumesi@academico.ufs.br]

Abstract. *The aim of this study is to compare the efficiency of machine learning algorithms in detecting DDoS attacks in containerized WordPress environments, with a focus on the REST API. To simulate realistic attack scenarios, the Locust tool was used to generate massive access to the WooCommerce REST API. The resulting traffic was captured and analyzed enabling the generation of real datasets. The tests showed that the Random Forest algorithm outperformed the others, achieving an accuracy and F1-score of 1.00 in balanced scenarios, while the other algorithms yielded significantly lower results.*

Resumo. *O objetivo deste trabalho é comparar a eficiência de algoritmos de aprendizado de máquina na detecção de ataques DDoS em ambientes WordPress conteinerizados, focando na API REST. Para simular cenários realistas de ataque foi utilizada a ferramenta Locust, que gerou acessos massivos à API REST do WooCommerce. O tráfego resultante foi capturado e analisado possibilitando a geração de conjuntos de dados reais. Os testes demonstraram que o algoritmo Random Forest apresentou desempenho superior, com acurácia e F1-score de 1.00 em cenários balanceados, enquanto os demais algoritmos obtiveram resultados significativamente inferiores.*

1. Introdução

Com o avanço da *internet*, o interesse de pessoas e empresas em criarem seus *sites* cresceu, principalmente visando trazer maior visibilidade aos seus negócios. Por outro lado, muitos usuários comuns passaram a utilizar a *internet* para criar *blogs* como meio de expressão pessoal. Com o tempo, ficou evidente que o gerenciamento de conteúdo web não é tarefa trivial. Assim, na década de 1990, surgiram os Sistemas de Gerenciamento de Conteúdo (*Content Management Systems* – CMS), com o objetivo de facilitar a criação e organização de *sites* [DialHost Internet, 2018].

Essas plataformas beneficiam tanto desenvolvedores experientes quanto usuários sem conhecimento em programação, ao possibilitar a criação, edição e publicação de conteúdos por meio de interfaces intuitivas e *plugins* que ampliam as funcionalidades do *site*. Um dos principais CMS disponíveis é o WordPress, onde é possível desenvolver temas, *plugins* e funcionalidades personalizadas através da API REST [Red Hat, 2023].

A integração com a API REST permite interagir com o banco de dados, possibilitando a criação, edição, exclusão e busca de conteúdos.

Apesar das facilidades oferecidas pelos sistemas de gerenciamento de conteúdo, *sites* desenvolvidos nessas plataformas têm se mostrado particularmente vulneráveis a ataques cibernéticos [Olhar Digital, 2019], tais como os ataques de *Distributed Denial of Service* (DDoS – Negação de Serviço Distribuído), que comprometem a estabilidade do ambiente e prejudicam a experiência do usuário final.

Tais ataques exploram vulnerabilidades na API REST, bem como na estrutura do WordPress, impactando negativamente a eficiência dos mecanismos de orquestração de contêineres. Entre os principais fatores que contribuem para essas vulnerabilidades estão o uso de versões desatualizadas da plataforma CMS, a adoção de senhas fracas por administradores e a instalação de *plugins* inseguros [Zorz, 2022].

Uma forma de aumentar a segurança do serviço de hospedagem de *websites* através de plataformas CMS é isolar cada cliente ou *website*. Esse isolamento pode ser feito por contêineres que operam no nível do sistema operacional, o que resulta em um menor consumo dos recursos de *hardware* disponíveis [Microsoft Azure, 2025]. Porém, esse isolamento pode ser insuficiente para garantir a segurança dos sistemas. Mais do que mitigar os riscos, é necessário buscar a antecipação de cenários que possam comprometer a integridade dos sistemas. Nesse contexto, o aprendizado de máquina tem se destacado como uma ferramenta promissora para identificação desses cenários [Cisco, 2025].

Mediante tal panorama, o objetivo deste estudo (que é um recorte de um Trabalho de Conclusão de Curso, do curso de Sistemas de Informação) é **comparar a eficiência de algoritmos de aprendizado de máquina na detecção de ataques DDoS em ambiente WordPress conteinerizado, focando na API REST**. Em virtude do crescimento de ataques cibernéticos e da crescente utilização do WordPress por diversas empresas, é de extrema importância entender o comportamento dessas aplicações frente a esses ataques, aspecto que justifica a realização de trabalhos como este.

2. Fundamentação do Trabalho

2.1 Aspectos conceituais

Este trabalho se fundamenta em 4 conceitos fundamentais para sua compreensão. O primeiro são os **contêineres**. Como discutido por Mendes e Duarte (2019), trata-se de uma forma de virtualização que encapsula cada processo, fornecendo apenas os arquivos e dependências necessários para sua execução. Por ser portátil, permite que múltiplas máquinas o executem a partir de uma mesma configuração, garantindo consistência e facilidade de implementação.

Entre os tipos de contêineres destaca-se o *Docker,* o qual recebeu esse nome em referência às pessoas que trabalham com carga e descarga de navios no Reino Unido, que são chamadas de *Dockers*. Desta forma, assim como *Dockers* no Reino Unido empacotam as cargas dos navios, a Tecnologia *Docker* "embala" aplicações de forma portável [Konrad, 2015]. O *Docker* possui uma maneira eficiente de executar e isolar aplicações através da combinação de camadas (Vitalino e Castro (2018).

O segundo fundamento é o **Sistema Gerenciador de Conteúdo Wordpress**. Dados divulgados pela Hostnet (2024) afirmam que o WordPress é o sistema de gerenciamento de conteúdo mais adotado no mundo. Essa poderosa ferramenta oferece uma série de vantagens, como a facilidade na criação de *sites* corporativos, portfólios, lojas virtuais e *blogs*. Uma das funcionalidades importantes do WordPress é a API REST que oferece diversas vantagens, como flexibilidade e a capacidade de criar integrações personalizadas, utilizando padrões pré-estabelecidos para criar, ler, atualizar e excluir dados. Tal recurso é utilizado para conectar aplicativos entre si, de forma simples e leve.

O terceiro conceito a ser abordado são os **Ataques Distribuídos de Negação de Serviço** (**DDoS**), os quais utilizam uma rede de dispositivos previamente comprometidos e espalhados geograficamente para gerar tráfego malicioso em larga escala contra alvos específicos. Esse tipo de ataque continua sendo uma ameaça significativa, impactando múltiplas camadas da rede, protocolos e serviços essenciais [Albano *et al*., 2023].

Por fim, o quarto fundamento é o **Aprendizado de Máquina**. Segundo Ferreira e Cavalcante (2019) a aprendizagem de máquina confere às máquinas a capacidade de resolver problemas por meio de algoritmos que aprendem com os dados e realizam previsões. No campo da cibersegurança, a capacidade de prever anomalias comportamentais é fundamental para bloquear atividades maliciosas em tempo real.

2.2 Trabalhos Relacionados

Foi realizada uma busca por trabalhos relacionados ao tema da pesquisa. Cada estudo foi examinado com o intuito de identificar as técnicas de aprendizado de máquina aplicadas, os ambientes de experimentação utilizados e os conjuntos de dados empregados no treinamento dos algoritmos. Em todos os casos, os ataques DDoS foram o foco, com exceção do trabalho de Lima Filho (2019) que contou também com os ataques DoS. O quadro 1 sintetiza tais características da literatura prévia encontrada.

Quadro 1. Síntese dos estudos anteriores

Autor	Técnica de Aprendizado	Ambiente	Conjunto de Dados
Lima Filho (2019)	Árvore de Decisão, Floresta Aleatória	Redes SDN	Dataset Autoral, CSECIC-IDS2018, CICIDS2017, ISCXIDS2012, CIC-DoS
Ferreira (2021)	K-NN, Árvores de Decisão, SVMs, Florestas Randômicas, Redes Neurais	Tráfego de Rede	Dataset Autoral, NSL-KDD, CICDDoS2019
Almeida Neto (2021)	BIRCH, Mini-batch, K-Means, Clustream, StreamKM++, DenStream, D-Stream	SDN/NFV	Dataset Autoral
Santos Neto (2021)	ISCXSlowDDoS2016, ISCXIDS2012, CTU10, CTU-11, ISOT	Redes SDN	Dataset Autoral

Nicola, Lauretto e Delgado (2021)	K-NN, Naive Bayes Random Forest, Regressão Logística, SVM	Google Colaboratory	Dataset fornecido pela ULB Machine Learning Group
Teles (2022)	Árvore de Decisão, KNN, Naive Bayes	Aplicação Web Local	Kaggle
Araújo (2023)	XGBoost	Tráfego de Rede	CIC-DDoS 2019
Chagas (2024)	XGBoost, LSTM, MLP, RN, RNR	SGBDs	Dataset de SGBD órgão da cúpula do Poder Judiciário Federal

Em síntese, 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. Contudo, o diferencial desta pesquisa em relação às anteriores é que foram aplicadas técnicas de aprendizado de máquina no contexto de contêineres e Wordpress.

Com base nos conceitos abordados e nos estudos analisados, segue-se para a seção de Metodologia, na qual se detalha como esses elementos foram integrados para viabilizar o alcance dos objetivos desta pesquisa.

3. Metodologia

Trata-se de uma pesquisa exploratória e qualitativa dividida em duas etapas. Na primeira fase, os dados utilizados para treinamento foram coletados no *dataset* CIC-DDoS2019.[1] O *dataset* foi criado pelo CIC, o Instituto Canadense de Segurança Cibernética, com o objetivo de incentivar a pesquisa e a educação no mundo da cibersegurança. Os algoritmos utilizados foram o *Random Fores*t (RF), Regressão Logística (RL) e *Support Vector Machine* (SVM). Foi conduzida uma série de etapas voltadas ao pré-processamento e tratamento dos dados, com o objetivo de viabilizar o treinamento dos modelos. A divisão dos dados seguiu a estratégia hold-out, com 70% para treinamento e 30% para teste, utilizando train_test_split da biblioteca scikit-learn. Nos cenários balanceados, aplicou-se oversampling para equilibrar as classes e reduzir viés nos modelos.

A segunda etapa consistiu na coleta de dados das chamadas à API REST. Para tal, foi desenvolvido um *site* simples com estrutura de comércio eletrônico no ambiente Wordpress, com o propósito de simular cenários reais de navegação e possibilitar a realização de testes de estresse e ataques do tipo DDoS, variando-se os volumes de requisições. O quadro 2 apresenta uma síntese das principais ferramentas adotadas.

Quadro 2. Apresentação das ferramentas adotadas.

Ferramenta	Objetivo
Locust	Simular ataques DDoS, emulando múltiplos usuários simultâneos acessando a API REST de forma intensiva.

[1] CIC-DDoS2019

tcpdump	O tráfego gerado durante as simulações foi capturado com a ferramenta tcpdump, que armazenava os pacotes em arquivos .pcap para posterior análise.
Wireshark	Utilizada para a análise do tráfego, permitindo a visualização detalhada do número de requisições, tempo de resposta da API REST e códigos de *status* HTTP retornados.
docker stats	Adotado para analisar o comportamento do ambiente containerizado que forneceu métricas em tempo real sobre o uso de CPU, memória e rede pelos contêineres envolvidos. Essas métricas foram coletadas durante os testes para avaliar a sobrecarga imposta pelo ataque em diferentes intensidades: 100, 200, 500 e 1000 usuários simultâneos. Após a coleta dos dados, foi realizado o pré-processamento, que incluiu a limpeza dos dados, remoção de valores inconsistentes e a extração de atributos relevantes para a detecção de ataques DDoS, como tempo de resposta, volume de tráfego e padrões nos pacotes.
scikit-learn	Os dados processados foram utilizados para treinar os algoritmos *Random Forest* (RF), Regressão Logística (RL) e *Support Vector Machine* (SVM). Para tal, foi utilizada a biblioteca *scikit-learn* em Python. Os modelos foram avaliados com dados de teste não vistos, a fim de verificar sua capacidade de generalização e desempenho.

Para os testes práticos, foram considerados dois cenários distintos: com e sem balanceamento de classes. No cenário sem balanceamento, usou-se o conjunto original, refletindo uma condição realista, porém com possível viés na classificação da classe minoritária, já que algoritmos supervisionados tendem a favorecer a classe majoritária. Já no cenário com balanceamento, foi aplicada a técnica de *oversampling*, por meio da duplicação de registros da classe menos representada com o intuito de igualar a quantidade de amostras entre classes. Essa abordagem foi feita para mitigar o viés dos algoritmos e ter uma avaliação mais equiparada do desempenho, especialmente na detecção de ataques.

As métricas utilizadas para avaliação incluíram acurácia, F1-score, taxa de detecção (*True Positives*), taxa de falsos positivos (*False Positives*) e tempo de resposta do modelo. Por fim, a comparação entre os resultados permitiu identificar qual algoritmo apresentou melhor desempenho na detecção de ataques DDoS direcionados à API REST do WooCommerce, possibilitando a reflexão sobre estratégias viáveis de mitigação em ambientes conteinerizados.

4. Resultados

Na primeira etapa deste trabalho, foram aplicados três algoritmos de aprendizado de máquina: *Random Forest*, Regressão Logística e Support Vector Machine (SVM) sob o conjunto de dados DDoS via UDP, que simula cenários de ataque de negação de serviço distribuído (DDoS), com e sem a aplicação de técnicas de balanceamento de classes. O conjunto é composto por dois arquivos CSV, correspondentes a capturas realizadas em dias distintos: 03 de novembro (03.11) e 01 de dezembro de 2024 (01.12).

O modelo *Random Forest* apresentou melhor desempenho em todos os cenários, alcançando métricas próximas a 1,00 mesmo sem a aplicação de técnicas de balanceamento, o que evidencia sua robustez diante do desbalanceamento de classes. Já Regressão Logística e o SVM foram impactados por conta da desproporção entre as classes. Sem balanceamento, eles apresentaram *recall* reduzido para a classe benigna, com destaque negativo para o SVM, onde o F1-score da classe zero foi inferior a 0,30 no pior caso.

Depois da aplicação de *oversampling*, esses modelos passaram a apresentar desempenho significativamente superior, com aumento expressivo no recall e no F1-score da classe zero. Esses resultados mostraram o quão é importante o balanceamento de classes na detecção de tráfego malicioso, principalmente quando se utilizam algoritmos mais sensíveis ao viés gerado por distribuições desiguais. O *Random Forest* se destacou como o algoritmo mais estável e confiável ao lidar com essa característica dos dados.

A Tabela 1 apresenta uma síntese dos resultados até agora discutidos. Para fins de explicação, o termo, "03.11" refere-se ao arquivo CSV-03-11.zip, com registros capturados em 3 de novembro, enquanto “01.12" refere-se ao conjunto de dados extraído do arquivo CSV-01-12.zip, correspondente às coletas realizadas em 1º de dezembro, conforme disponibilizado no repositório oficial do CICDDoS2019. Nestes arquivos “zip” estão agregados vários tipos de ataques, sendo adotado o DrDoS_UDP.csv, que é um tipo de ataque UDP Flood o qual é simples e eficiente na sobrecarga de largura de banda, sendo frequente em ataques reais.

Tabela 1. Resultados da comparação entre o Random Forest (RF), Support Vector Machine (SVM) e Regressão Logística (RL) com a base do CIC DDoS.

Métricas	**Sem balanceamento**						**Com balanceamento**					
	DrDoS UDP 03.11			**DrDoS UDP 01.12**			**DrDoS UDP 03.11**			**DrDoS UDP 01.12**		
	RF	**SVM**	**RL**	**RF**	**SVM**	**RL**	**RL**	**SVM**	**RL**	**RL**	**SVM**	**RL**
Amostras Classe 0 (Benigno)	606.0	948	948	948	606	29971	3017	3017	1539	3017	3017	1539
Amostras Classe 1 (Ataque)	1110600	1110600	1110600	928.233	928233	928230	2983	2983	1461	2983	2983	1461
Acurácia	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.94	0.98	1.00	0.94	0.98
Precisão (Classe 0)	0.99	0.73	0.97	1.00	0.40	0.98	1.00	0.89	1.00	1.00	0.89	1.00
Recall (Classe 0)	1.00	0.67	0.77	0.99	0.20	0.40	1.00	0.99	0.96	1.00	0.99	0.96
F1-score (Classe 0)	1.00	0.70	0.86	0.99	0.26	0.56	1.00	0.94	0.98	1.00	0.94	0.98
Precisão (Classe 1)	1.00	1.00	1.00	1.00	1.00	0.98	1.00	0.99	0.96	1.00	0.99	0.96
Recall (Classe 1)	1.00	1.00	1.00	1.00	1.00	1.00	1.00	0.88	1.00	1.00	0.88	1.00
F1-score (Classe 1)	1.00	1.00	1.00	1.00	1.00	0.99	1.00	0.93	0.98	1.00	0.93	0.98

Macro Avg – F1-score	1.00	0.85	0.93	1.00	0.63	0.78	1.00	0.94	0.98	1.00	0.94	0.98

Na segunda etapa foi configurado um ambiente com a plataforma WordPress em contêiner *Docker*, com a API REST do WooCommerce e um *script* foi projetado para ser facilmente ajustado com diferentes quantidades. Para avaliar a capacidade de generalização do modelo treinado, realizou-se um experimento com dados reais obtidos a partir da simulação de um ataque DDoS via UDP com 10.000 usuários simultâneos direcionados à API REST do WooCommerce. O tráfego foi processado com o CICFlowMeter, garantindo compatibilidade com os *datasets* anteriores. Como o conjunto real gerado continha apenas fluxos maliciosos (classe 1) foi aplicado um *dataset* balanceado artificialmente, duplicando registros de ataque e rotulando-os como benignos. Nesse cenário, os algoritmos Regressão Logística e SVM apresentaram desempenho significativamente inferior, com F1-score médio abaixo de 0,50. Isso se deve à ausência de distinção real entre as classes, o que inviabilizou a aprendizagem de padrões discriminativos. Já o *Random Forest* destacou-se por sua robustez, atingindo desempenho perfeito (F1 = 1,00) mesmo nesse cenário artificialmente balanceado, demonstrando maior capacidade de adaptação mesmo em condições adversas.

O experimento também revelou que, embora o ambiente em contêiner *Docker* tenha se mantido estável durante os testes com Locust, a API apresentou alta latência e respostas com erro 500 sob carga extrema mostrando que ataques DDoS ainda comprometem a disponibilidade, mesmo em infraestruturas conteinerizadas. Assim, ficou evidente o potencial do *Random Forest* como estratégia viável para detecção precoce de tráfego malicioso em tempo real. Os resultados aqui discutidos são organizados na Tabela 2 para uma visualização mais completa dos dados.

Tabela 2. Resultados da comparação entre o Random Forest (RF), Support Vector Machine (SVM) e Regressão Logística (RL) no ambiente Wordpress.

Origem dos dados: Wordpress			
Métricas	**Com balanceamento**		
	RF	**SVM**	**RL**
Amostras Classe 1 (Ataque)	6520	6520	6520
Amostras Classe 0 (Benigno)	6511	6511	6511
Acurácia	1.00	0.48	0.49
Precisão Classe 0	1.00	0.48	0.48
Precisão Classe 1	1.00	0.49	0.49
Recall Classe 0	1.00	0.47	0.42
Recall Classe 1	1.00	0.50	0.55
F1-Score Classe 1	1.00	0.49	0.52

Weighted Avg – F1-score	1.00	0.48	0.48

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.

De forma semelhante, na detecção dos ataques DrDoS via UDP, o *Random Forest* manteve alto desempenho mesmo sob condições adversas de distribuição de classes, o que mostra sua capacidade de generalização. Essa consistência contrasta com modelos mais sensíveis ao desbalanceamento, como a Regressão Logística e o SVM, os quais apresentaram desempenho significativamente inferior no cenário com dados reais simulados e balanceamento artificial.

5. Conclusão

Os resultados mostraram que o uso de algoritmos de aprendizado de máquina pode ser uma abordagem eficaz na detecção de ataques DDoS direcionados a ambientes WordPress conteinerizados. Dentre os algoritmos avaliados o modelo *Random Forest* apresentou desempenho superior em todos os cenários, com precisão e F1-score acima de 0.99, mesmo quando aplicado a dados balanceados artificialmente e a conjuntos gerados a partir de tráfego real.

Esse destaque se deve à capacidade do *Random Forest* de lidar com padrões complexos e não lineares, além de sua robustez frente à presença de ruído ou dados redundantes — características comuns em fluxos de rede sob ataque. Já os modelos lineares SVM e Regressão Logística apresentaram desempenho mais limitado, com F1-scores próximos de 0.48, sugerindo maior sensibilidade à simetria artificial dos dados e à ausência de variações mais ricas no tráfego benigno.

Diante disso, conclui-se que a aplicação de modelos de aprendizado de máquina, especialmente algoritmos baseados em árvores como o *Random Forest*, representa uma solução promissora para a detecção precoce de DDoS em APIs de serviços *web*. A integração dessa detecção com monitoramento contínuo e respostas automatizadas pode ampliar significativamente a resiliência de plataformas como o WordPress, mesmo quando executadas em infraestrutura conteinerizada.

Como trabalho futuro, pretende-se realizar testes comparativos entre ambientes conteinerizados e ambientes virtualizados com máquinas virtuais (VMs). Essa comparação poderá revelar diferenças relevantes no gerenciamento de recursos, resposta a sobrecargas e tolerância a ataques de negação de serviço, uma vez que VMs possuem um grau de isolamento maior do que contêineres, ao custo de maior consumo de recursos. Tal análise pode contribuir para a definição mais precisa da infraestrutura mais resiliente para aplicações WordPress expostas à internet.

Além disso, recomenda-se a ampliação dos testes com algoritmos de aprendizado de máquina mais avançados, como o XGBoost ou LightGBM, que vêm

demonstrando resultados superiores em diversas competições de ciência de dados e cenários de detecção de anomalias. A comparação entre esses algoritmos e o *Random Forest* pode oferecer ganhos em desempenho, além de revelar potenciais vantagens em termos de velocidade de inferência, uso de memória ou generalização. Essa investigação contribuiria para refinar ainda mais os mecanismos de defesa baseados em aprendizado de máquina contra ataques DDoS.

Referências

Albano, L.; Borges, L.F.; Neira, A.B. and Nogueira, M. (2023) Predição de Ataques DDoS pela Correlação de Séries Temporais via Padrões Ordinais. In: XXIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais, 1-14. Disponível em: https://sol.sbc.org.br/index.php/sbseg/article/view/27198

Almeida Neto, J.R. (2021) Detecção de ataques DDoS em ambientes SDN/NFV utilizando algoritmos de aprendizagem de máquina não supervisionados em fluxos de dados. Dissertação (Mestrado em Ciência da Computação). Universidade Federal de Sergipe. Disponível em: https://ri.ufs.br/handle/riufs/15022

Araújo, P. (2023) Impacto de métodos de seleção de variáveis na classificação de ataques DDoS utilizando XGBoost. Dissertação (Mestrado em Ciências) Universidade de São Paulo: https://www.teses.usp.br/teses/disponiveis/3/3142/tde-21092023-082915/pt-br.php

Chagas, D. M. Detecção de Ataques de Negação de Serviço em SGBDs A Partir de Logs Internos Usando Abordagens Supervisionada e Não Supervisionada. Dissertação (Mestrado em Engenharia Elétrica). Universidade de Brasília. Disponível em: https://repositorio.unb.br/handle/10482/48503

Cisco (2025) What Is Machine Learning in Security? Acesso em: 25 mar. 2025. Disponível em: https://www.cisco.com/c/en/us/products/security/machine-learning-security.html

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

Ferreira, A. and Cavalcante R. (2019) Análise comparativa entre algoritmos de aprendizagem de máquina em classificação de imagens de radiografia no auxílio ao diagnóstico de pneumonia. In: XIX Escola Regional de Computação Bahia, Alagoas e Sergipe, 1-10. Disponível em: https://sol.sbc.org.br/index.php/erbase/article/view/8985

Ferreira, M. (2021) Detecção de DDoS por aprendizado de máquina. Monografia (Graduação em Engenharia da Computação). Universidade de Brasília. Disponível em: https://bdm.unb.br/bitstream/10483/29831/1/2021_MatheusSiadeFerreira_tcc.pdf

Hostnet. Por que escolher o WordPress como a plataforma do seu site? 2024. Acesso em: 25 mar. 2025. Disponível em:

https://www.hostnet.com.br/blog/por-que-escolher-o-wordpress-como-a-plataforma-do-seu-site/

Konrad, A (2015) Meet Docker founder Solomon Hykes. Acesso em: 18 de dez. 2024. Disponível em: https://www.forbes.com/sites/alexkonrad/2015/07/01/meet-docker-founder-solomon-hykes/

Lima Filho, F.S. (2019) Smart Defender: um sistema de detecção e mitigação de ataques DoS/DDoS usando aprendizagem de máquina. Tese (Doutorado em Engenharia Elétrica e de Computação). Universidade Federal do Rio Grande do Norte. Disponível em: https://repositorio.ufrn.br/items/483539cb-44a4-4f94-810b-b9d57b39c465

Mendes, A. and Duarte, (2019) A. Comparativo entre DOCKER e LXC/LXD para a virtualização. In: XIX Escola Regional de Computação Bahia, Alagoas e Sergipe, 1-9. Disponível em: https://sol.sbc.org.br/index.php/erbase/article/view/8990

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

Nicola, V. G. O. M.; Lauretto, M. S. and Delgado, K. V. (2021) Avaliação empírica de classificadores e métodos de balanceamento para detecção de fraudes em transações com cartões de crédito. In: XVII Encontro Nacional de Inteligência Artificial e Computacional, 1-12. Disponível em: https://sol.sbc.org.br/index.php/eniac/article/view/12118/11983

Olhar Digital. (2019) Estudo afirma: de todos os sites CMS hackeados em 2018, 90% eram WordPress. Acesso em: 25 mar. 2025. Disponível em: https://olhardigital.com.br/2019/03/05/seguranca/estudo-afirma-de-todos-os-sites-cms-hackeados-em-2018-90-eram-wordpress /

Red Hat. (2023) O que é uma API REST? Acesso em: 25 mar. 2025. Disponível em: https://www.redhat.com/pt-br/topics/api/what-is-a-rest-api

Santos Neto, M.J. (2021). Detecção de ataque DDoS em SDN utilizando entropia e machine learning. Dissertação (Mestrado Profissional em Computação Aplicada). Disponível em: http://repositorio2.unb.br/handle/10482/40991?locale=en

Teles, J. G. N. (2022) Detecção de ameaças DDoS com aprendizagem de máquina. Monografia (Graduação em Engenharia Eletrônica e Telecomunicações). Universidade Federal de Uberlândia. Disponível em: https://repositorio.ufu.br/bitstream/123456789/34622/1/DeteccaoAmeacasDDoS.pdf

Vitalino, C. and Castro, P. (2018) Descomplicando o Docker. Brasil: BRASPORT,

Zorz, Z. (2022) CMS-based sites under attack: The latest threats and trends. Acesso em: 25 mar. 2025. Disponível em: https://www.helpnetsecurity.com/2022/05/03/cms-threats-trends/