Um sensor baseado em Aprendizado de Máquina para detecção de ataque DDoS em tempo real

M.A. Ribeiro[1], **M. Fonseca**[1], **J. Santi**[3]

[1]Programa de Pós-Graduação em Engenharia Elétrica e Informática Industrial (UTFPR)

[2]Departamento Acadêmico de Informática (UTFPR)

{marcos.ti.ribeiro}@gmail.com,{maurofonseca,jsanti}@utfpr.edu.br

Abstract. *Distributed denial of service (DDoS) aims to coordinate a synchronized attack on online systems using infected equipment (bots), causing slowness or unavailability of the service. Recently, this type of attack has evolved in intensity, diversity and economic impact. Within this context, this work aims to present a real-time DDoS detection tool based on a sensor that uses Machine Learning algorithms. A testing environment was developed to validate the tool's effectiveness. The performance and results of the different classifiers used in the sensor implementation will be discussed. The results indicate that the sensor is efficient in detecting DDoS attacks in approximately 3 seconds.*

Resumo. *A negação de serviço distribuído (DDoS) tem como objetivo coordenar um ataque sincronizado a sistemas online utilizando equipamentos infectados (bots), causando lentidão ou indisponibilidade do serviço. Recentemente, este tipo de ataque evoluiu em termos de intensidade, diversidade e impacto econômico. Dentro deste contexto, este trabalho tem como objetivo apresentar uma ferramenta de detecção de DDoS em tempo real com base em um sensor que utiliza algoritmos de Aprendizado de Máquina. Um ambiente de testes foi desenvolvido para validar a eficácia da ferramenta. Serão discutidos o desempenho e os resultados dos diferentes classificadores utilizados na implementação do sensor. Os resultados indicam que o sensor é eficiente na detecção do ataques DDoS em aproximadamente 3 segundos.*

1. Introdução

Arquiteturas computacionais vulneráveis resultam de projetos equivocados, falhas no desenvolvimento ou sistemas legados e, consequentemente, são gerados ecossistemas de tecnologias frágeis, permitindo que usuários mal-intencionados comprometam informações ou serviços [Jajodia et al. 2011]. Um dos ataques que exploram a fragilidade desses sistemas é o ataque de Negação de Serviço Distribuído comumente chamado DDoS (*Distributed Denial of Service*). Impulsionado pela crescente demanda de serviços *online*, o ataque DDoS tornou-se mais efetivo e destrutivo, dado que um ataque direcionado pode afetar milhares de usuários [Cil et al. 2021].

Segundo [Nanda et al. 2016], através do fluxo padrão do TCP/IP e das características de navegação de usuários mal-intencionados é possível diferenciar entre usuários atacantes e usuários legítimos. O Aprendizado de Máquina (*Machine Learning* - ML) pode ser usado para identificar esses padrões. O ML é utilizado essencialmente para

cenários em que seja necessário previsão de resultados e predição a partir de padrões e inferências. Essa abordagem tem apresentado resultados expressivos na classificação de fluxo de usuários nas redes de computadores.

Considerando o contexto exposto, esse trabalho apresenta um sensor com arquitetura modular e flexível para detecção de ataques DDoS em tempo real. O sensor aplica algoritmos de Aprendizado de Máquina sobre fluxos coletados, classificando-os em fluxos benignos ou maliciosos. A validação e avaliação da solução foi realizada através de simulação dentro de uma arquitetura SDN (*Software Defined Networking*).

Este artigo está organizado da seguinte forma: na Seção 2 apresentam-se os trabalhos relacionados. A Seção 3 aborda o referencial teórico dos principais temas presentes no artigo. A Seção 4 descreve a arquitetura do projeto e detalha as implementações desenvolvidas. A Seção 5 apresenta os resultados da avaliação da arquitetura proposta. O artigo é concluído na Seção 6.

2. Trabalhos Relacionados

Existem diversos estudos de detecção de ataques DDoS documentados na literatura. Em [Sahoo et al. 2018] foram avaliados os algoritmos *k-Nearest Neighbor*, *Naive Bayes*, *Decision Tree*, *Support Vector Machine*, *Random Forest* e *Linear Regression*. Entretanto, a solução de detecção do projeto não é independente do projeto desenvolvido e não permite sua refatoração para projetos alternativos.

[Barki et al. 2016] apresenta um *Intrusion Detection System* (IDS) com ACL (*Access Control List*) que, através de algoritmos de ML, permite acesso/bloqueio de determinado recurso. Foram utilizados os algoritmos *k-Nearest Neighbor*, *Naive Bayes* e *k-Mens*. O presente projeto expandiu as opções de escolha de algoritmos de ML, permitindo ao administrador a seleção do algoritmo conforme a política adotada pela organização (por exemplo, é possível escolher o modelo que permite detectar melhor falsos negativos).

Diferentemente dos estudos citados, este trabalho apresenta uma ferramenta capaz de detectar ataque DDoS e permitir que outras ferramentas utilizem seus resultados para a mitigação de ataques DDoS dentro de suas plataformas. A intuito é utilizar a capacidade de adaptação e flexibilidade da ferramenta para integração com outras tecnologias.

3. Referencial teórico

Nesta seção, apresentamos os conceitos utilizados neste trabalho.

3.1. Ataque DDoS

O ataque DDoS é definido como o esforço conjunto de vários dispositivos para ocasionar interrupção de um determinado serviço da Internet. A fase inicial do ataque se concentra no sequestro de dispositivos distribuídos na Internet através de infecção de *malwares* ou vírus. Estes dispositivos são organizados para orquestrar um ataque, formando uma rede de *bots* (*Botnet*). A *Botnet* permite que os invasores atualizem seus métodos de invasão, uma vez que possuem acesso remoto as máquinas infectadas, o que torna a detecção e o bloqueio por origem ineficazes. Os ataques, que são iniciados através de instruções enviadas aos dispositivos contaminados [Maheshwari et al. 2022], são constantemente atualizados na medida em que dispositivos de segurança identificam as caraterísticas e ferramentas de ataque.

3.2. Aprendizagem de máquina

O principal desafio no diagnóstico do ataque DDoS é diferenciar o tráfego gerado pelos ataques do tráfego legítimo. Essa tarefa se torna mais complexa pela quantidade de tráfego majoritariamente legítimo que transpassa a rede [Zhou et al. 2022]. Esses foram os principais motivadores das pesquisas utilizando *Machine Learning* nas redes de computadores [Dayal and Srivastava 2018].

O Aprendizado de Máquina é um método de análise de dados que automatiza a construção de modelos analíticos, usando algoritmos que possuem a capacidade de aprender interativamente a partir de dados coletados. Para permitir que o algoritmo aprenda é necessário um de conjunto de dados (*dataset*) preciso, para gerar a classificação de tráfego normal em relação ao tráfego gerado pelo ataque [Dayal and Srivastava 2018].

Neste trabalho são considerados quatro modelos de classificadores de famílias distintas: modelos probabilísticos (*Gaussian Naive Bayes* - GNB), lineares (*Support Vector Machine* - SVM), árvores (*Random Forest* - RF) e rede neurais artificiais (*Multilayer Perceptron* - MLP). Um *ensamble* Pilha de Classificadores (PC) destes algoritmos também é usado como sensor. Para detalhamento sobre os classificadores, consultar a página do projeto https://marcostiribeiro.github.io/toolroom e as pesquisas de [Occhipinti et al. 2022, Ganaie et al. 2022].

4. Arquitetura do Sensor

Nesta seção, apresenta-se a arquitetura do sensor para detecção de ataques DDoS. O sensor é responsável por realizar a classificação dos dados e notificar o servidor de segurança caso um cliente seja classificado como possível ameaça. O sensor foi desenvolvido na linguagem Python 3.10, utilizando diversas técnicas e ferramentas que foram incorporadas ao fluxo interno da aplicação. O sensor é dividido em dois módulos: módulo de treinamento e módulo de detecção. Os códigos fontes do projeto e a documentação estão disponíveis em https://marcostiribeiro.github.io/toolroom/. O vídeo de demonstração da ferramenta está disponível em https://youtu.be/4J1sVA6AjIA.

4.1. Módulo de Treinamento

O módulo é responsável pelo treinamento do modelo de Aprendizado de Máquina. O *dataset* utilizado no treinamento é resultado das pesquisas de [Sharafaldin et al. 2019] denominado CICDDOS2019. Os dados disponibilizados foram compilados em um conjunto de dados único, sendo utilizados (12.794.627) registros, balanceados em fluxos benignos e maliciosos (DDoS). Inicialmente o *dataset* é carregado. Em seguida é realizado o Pré-Processamento sobre os dados carregados e, na sequência, os modelos são treinados. O modelo proposto foi treinado com os algoritmos citados na Seção 3.2. Os modelos são treinados individualmente e são armazenados para serem utilizados durante a detecção, minimizando o tempo de treinamento durante a execução do sensor.

4.2. Módulo de Detecção

O módulo é responsável pela detecção do fluxo malicioso e seu funcionamento é detalhado na Figura 1. No início de sua execução é selecionado o modelo que atuará na instância de execução do sensor. Um requisito opcional de segurança é a criação de senha para o sensor no servidor de segurança (Fig. 1 - Solicita Permissão Server). Neste

caso, o sensor é configurado previamente no servidor de segurança com *login* (ID do sensor), senha e *Mac-address*. Na sequência o sensor solicita ao servidor de segurança parâmetros para realizar a comunicação entre o sensor e o servidor de segurança. Durante a solicitação, o sensor solicitará autenticação remota, utilizando os parâmetros previamente configurados (*login*, senha e *Mac-Address*). A validação dos parâmetros é realizado pelo servidor de segurança.

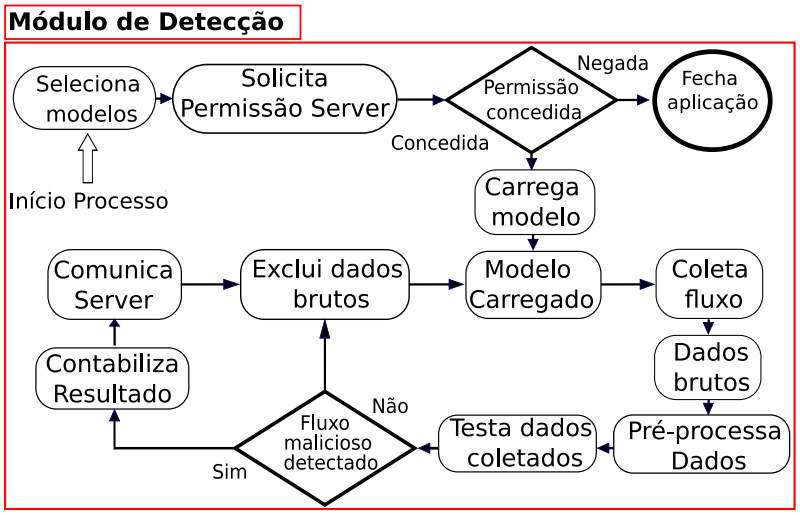

Figura 1. Diagrama do módulo de detecção.

Com permissão concedida é carregado o modelo de algoritmos de Aprendizado de Máquina armazenado durante a fase de treinamento. Os modelos são usados para classificar o fluxo de dados como benigno ou malicioso (Seção 3.2). A coleta de dados (Fig. 1 - Coleta fluxo) é realizada através da ferramenta Tcpdump [1] através do espelhamento de porta *default gateway*do *switch*.

A partir dos dados coletados, as caraterísticas existentes no conjunto de dados de treino são pré-processadas utilizando a iniciativa CICFlowMeter [Draper-Gil et al. 2016, Lashkari et al. 2017]. Os dados pré-processados são entregues aos algoritmos de Aprendizado de Máquina para análise. Os sensor pode atuar de dois modos: resumido e probabilístico. No modo resumido, se o fluxo for classificado como malicioso o sensor enviará uma mensagem ao servidor. Na mensagem enviada constará IP do usuário suspeito para a inclusão na lista de *hosts* maliciosos. No modo probabilístico, o sensor envia o IP de todos os servidores selecionados juntamente com a probabilidade atribuída a cada um deles, ficando a cargo da ferramenta de segurança a sensibilidade da detecção do ataque. Para reduzir a carga de processamento no servidor de segurança e o tráfego da rede de computadores, os usuários diagnosticados como maliciosos são contabilizados no sensor (Fig. 1 - Contabiliza Resultado). Um relatório é gerado com o IP, quantidade de ataques e data/hora do detecção.

O sensor também possui a característica de detecção de ataque local. Neste caso, conforme as configurações realizadas, o sensor realizará o bloqueio do IP de origem utilizando o *firewall* local do sistema operacional.

[1] https://www.tcpdump.org/ - acesso em 10 de março, 2024

5. Avaliação de desempenho

Para avaliar o desempenho do sensor foram realizados experimentos baseados em simulação. Para a simulação foi desenvolvido um ambiente SDN (*Software Defined Networking*) [Yungaicela-Naula et al. 2022], caso o sensor detecte um cliente malicioso o controlador do sistema SDN será comunicado. Nesse experimento, o controlador SDN foi utilizado como servidor de segurança. Detalhes sobre a evolução do ambiente de teste pode ser encontrado em https://marcostiribeiro.github.io/toolroom/demonstration.html

Para os experimentos foi utilizado um computador com Core i7 4.90 GHz com 32Gb de memória e sistema operacional Linux 5.0.0-38. A topologia de rede (Figura 2) foi criada utilizando a ferramenta para emular rede GNS3 [2], utilizando máquinas virtuais.

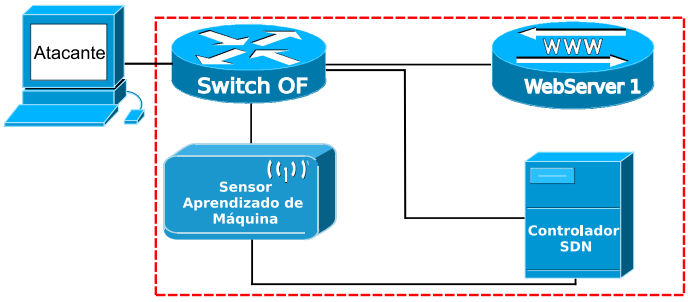

Figura 2. Topologia usada na avaliação de desempenho.

Os resultados da avaliação são apresentados considerando i) o comparativo de desempenho entre diferentes algoritmos de ML usados (Seção 5.1), e ii) a capacidade de detecção do ataques DDoS, observando-se os tempos de resposta (Seção 5.2).

5.1. Desempenho dos algoritmos de Aprendizado de Máquina

Para avaliar o desempenho dos algoritmos de ML utilizados nos sensores (Seção 3.2) foram usadas três métricas calculadas a partir da matriz de confusão: acurácia, precisão e revocação [Dey et al. 2018].

A validação cruzada [Occhipinti et al. 2022] foi a técnica utilizada para avaliar a eficácia dos modelos preditivos usados no sensor proposto neste trabalho. O Figura 3 apresenta os resultados de acurácia, precisão e revocação para os algoritmos usados.

O algoritmo que conseguiu o melhor resultado para acurácia foi o Pilha de Classificadores (*ensamble*) com 93,31% seguido pelo classificador *Random Forest* com 92,36%. Os algoritmos *Support Vector Machine*, *Multilayer Perceptron* e *Gaussian Naive Bayes* obtiveram, respectivamente, 78,57%, 68,69% e 62,42% de acurácia.

Para a precisão, os algoritmos que obtiveram os piores resultados foram *Gaussian Naive Bayes* com 57,90% e o *Multilayer Perceptron* com 70,07%. O *Support Vector Machine* obteve 91,5%, o Pilha de Classificadores alcançou 96,43% e *Random Forest* com 98,62% com o melhor desempenho.

Considerando a revocação, o Pilha de Classificadores atingiu 90,11% nessa métrica, melhor que o classificador *Random Forest* que obteve 85,81%. Os que obtiveram

[2]https://gns3.com/ - acesso em 10 de março, 2024

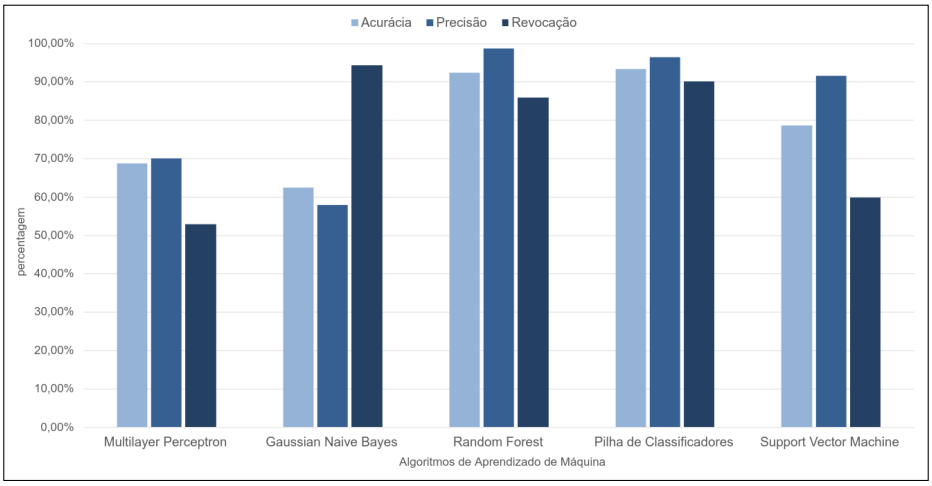

Figura 3. Resultado das métricas dos algoritmos de Machine Learning.

os piores resultados foram os classificadores *Multilayer Perceptron* (52,78%) e *Support Vector Machine* (59,8%). O *Gaussian Naive Bayes* teve resultado melhor entre todos os classificadores com 94,31% de revocação.

5.2. Detecção de ataque DDoS

Para gerar ataques baseados em fluxo, em protocolos e na camada de aplicação foram usados as ferramentas, Slowhttptest [3], Scapy [4] e Hping [5]. Os ataques DDoS considerados foram: *Bad TCP flags*, *FIN Only Set*, *SYN and FIN Set* e *TCP SYN Flood*.

O tempo de resposta da arquitetura proposta foi aferido de forma a verificar a eficiência do sensor em detectar o ataque e informar ao controlador SDN. O tempo médio para cada sequência de teste é definido conforme a Equação (1):

$$\overline{x} = \frac{1}{N} \sum_{n=1}^{N} (mitigTime - initAttack) \tag{1}$$

onde N=20 é número de amostras coletadas para o ataque, *initAttack* é o instante de tempo em que o ataque é iniciado, *mitigTime* é o instante de tempo em que o ataque é mitigado, e $\overline{x}$ é a média de tempo dos ataques realizados. O tempo médio de resposta (em segundos) do sensor e algoritmos ML (Seção 3.2), é apresentado na Figura 4.

O tempo médio para detecção de ataques considerando todos os algoritmos foi 2,8660 segundos (Figura 4). O nível de confiança usado é de 95% para os resultados apresentados. O modelo Pilha de Classificadores foi o que apresentou maior velocidade na detecção de ataques (tempo médio de resposta de 2,8263 segundos), seguido por *Gaussian Naive Bayes* (2,8594 segundos), *Support Vector Machine* (2,8597 segundos), *Multilayer Perceptron* (2,8797 segundos) e *Random Forest* (2,9047 segundos). A proximidade nos tempos de resposta gerados mostra que a utilização de algoritmos de famílias distintas não interfere na agilidade da classificação dos sensores.

[3]https://github.com/shekyan/slowhttptest/ - acesso em 10 de março, 2024
[4]https://scapy.net/ - acesso em 10 de março, 2024
[5]http://www.hping.org/ - acesso em 10 de março, 2024

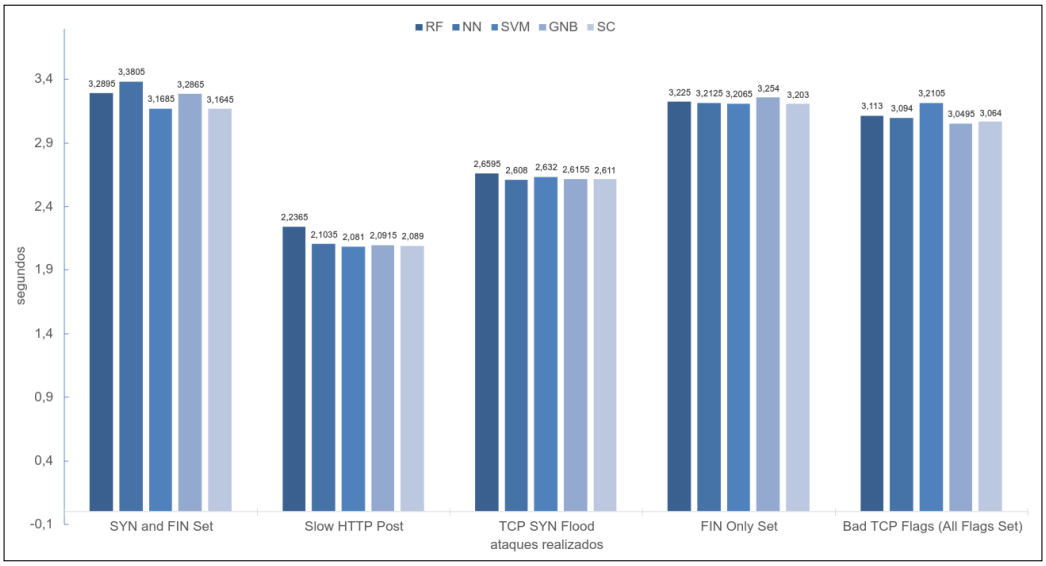

Figura 4. Tempo médio de resposta (em segundos) dos sensores com diferentes classificadores.

Em relação à detecção dos diferentes tipos de ataques considerados, o modelo proposto apresentou os menores tempos de resposta na detecção de ataques *Slow* HTTP *Post* e TCP SYN *Flood* (Figura 4). Os ataques baseados em inundação (por exemplo, o TCP SYN *Flood*) são os tipos mais frequente de ataque DDoS utilizado contra grandes companhias e o que possui maior efetividade [Cil et al. 2021]. Por sua vez, o ataque *Slow* HTTP *Post* é de difícil diagnóstico devido sua semelhança com fluxos de usuários legítimos [Hong et al. 2017]. Assim, os resultados gerados evidenciam a capacidade e agilidade do modelo proposto em detectar e mitigar ataques de alta incidência e de difícil detecção.

6. Conclusão e trabalhos futuros

Este trabalho apresenta uma ferramenta modular para detectar ataques DDoS em tempo real utilizando Aprendizado de Máquina. A modularização e capacidade de adaptação das tecnologias empregadas proporcionam flexibilidade, inclusive para lidar com outros tipos de ataques de segurança. Os resultados dos experimentos demonstram que o sensor é rápido e eficiente na classificação de fluxos, independentemente do modelo preditivo utilizado e do tipo de ataque realizado. O sensor é flexível, permitindo que seja utilizado apenas um classificador, um conjunto de classificadores (*ensemble*) ou outros métodos de classificação diferentes dos empregados neste trabalho. Essa decisão fica a cargo do administrador da rede, que pode ajustar a sensibilidade dos modelos para o diagnóstico de ataques, o que impacta diretamente na gestão dos recursos utilizados pelo sensor. Futuramente, pretende-se otimizar os modelos e expandir os testes para diferentes ambientes e paradigmas, como, por exemplo, a borda da rede (*network edge*) e o núcleo da rede (*network core*).

Referências

Barki, L., Shidling, A., Meti, N., Narayan, D., and Mulla, M. M. (2016). Detection of distributed denial of service attacks in software defined networks. In *Int. Conf. on Advances in Computing, Communications and Informatics*, pages 2576–2581.

Cil, A. E., Yildiz, K., and Buldu, A. (2021). Detection of ddos attacks with feed forward based deep neural network model. *Expert Systems with Applications*, 169:114520.

Dayal, N. and Srivastava, S. (2018). An rbf-pso based approach for early detection of ddos attacks in sdn. In *Int. Conf. on Communication Systems & Networks*, pages 17–24.

Dey, S. K., Rahman, M. M., and Uddin, M. R. (2018). Detection of flow based anomaly in openflow controller: Machine learning approach in software defined networking. In *Int. Conf. Electrical Engineering and Information Com. Technology*, pages 416–421.

Draper-Gil, G., Lashkari, A. H., Mamun, M. S. I., and Ghorbani, A. A. (2016). Characterization of encrypted and VPN traffic using time-related features. In *International Conference on Information Systems Security and Privacy*, pages 407–414. SciTePress.

Ganaie, M., Hu, M., Malik, A., Tanveer, M., and Suganthan, P. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, 115:105151.

Hong, K., Kim, Y., Choi, H., and Park, J. (2017). Sdn-assisted slow http ddos attack defense method. *IEEE Communications Letters*, 22(4):688–691.

Jajodia, S., Ghosh, A. K., Swarup, V., Wang, C., and Wang, X. S. (2011). *Moving target defense: creating asymmetric uncertainty for cyber threats*, volume 54. Springer Science & Business Media.

Lashkari, A. H., Draper-Gil, G., Mamun, M. S. I., and Ghorbani, A. A. (2017). Characterization of Tor traffic using time based features. In *International Conference on Information Systems Security and Privacy*, pages 253–262. SciTePress.

Maheshwari, A., Mehraj, B., Khan, M. S., and Idrisi, M. S. (2022). An optimized weighted voting based ensemble model for ddos attack detection and mitigation in sdn environment. *Microprocessors and Microsystems*, 89:104412.

Nanda, S., Zafari, F., DeCusatis, C., Wedaa, E., and Yang, B. (2016). Predicting network attack patterns in sdn using machine learning approach. In *IEEE Conf. on Network Function Virtualization and Software Defined Networks*, pages 167–172. IEEE.

Occhipinti, A., Rogers, L., and Angione, C. (2022). A pipeline and comparative study of 12 machine learning models for text classification. *Expert Systems with Applications*, 201:117193.

Sahoo, K. S., Iqbal, A., Maiti, P., and Sahoo, B. (2018). A machine learning approach for predicting ddos traffic in software defined networks. In *2018 International Conference on Information Technology (ICIT)*, pages 199–203.

Sharafaldin, I., Lashkari, A. H., Hakak, S., and Ghorbani, A. A. (2019). Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In *2019 International Carnahan Conference on Security Technology (ICCST)*, pages 1–8.

Yungaicela-Naula, N. M., Vargas-Rosales, C., Pérez-Díaz, J. A., and Zareei, M. (2022). Towards security automation in software defined networks. *Computer Communications*, 183:64–82.

Zhou, L., Zhu, Y., Zong, T., and Xiang, Y. (2022). A feature selection-based method for ddos attack flow classification. *Future Generation Computer Systems*, 132:67–79.