Atualização Confiável dos Modelos de Detecção de Intrusão Baseada em Aprendizagem de Máquina

Pedro Horchulhack[1], **Altair Olivo Santin**[1], **Eduardo Kugler Viegas**[1]

[1]Programa de Pós-Graduação em Informática (PPGIa)
Pontifícia Universidade Católica do Paraná (PUCPR)
80.215-901 – Curitiba – PR

{pedro.horchulhack, santin, eduardo.viegas}@ppgia.pucpr.br

Abstract. *This work proposes a new method for updating intrusion detection models using stream learning, reducing the instances needed for updates and computational costs. Rejected instances are stored for incremental updates, allowing automatic labeling from public repositories. Experiments with a 2.6TB database showed that the proposal maintains high accuracy for up to three months, reducing false positives by up to 12% and rejecting 8% of instances. Periodic updates improve accuracy by up to 6%, consuming only 3.2% of processing time and 2% of new instances compared to traditional techniques.*

Resumo. *Este trabalho apresenta um novo método para atualizar modelos de detecção de intrusão usando aprendizado de fluxo, reduzindo eventos para atualização e custos computacionais. Instâncias rejeitadas na classificação são armazenadas para atualização incremental, permitindo rotulação automática a partir de repositórios públicos. Experimentos mostraram que a proposta reduz os falsos-positivos em até 12%, rejeitando 8% das instâncias, em uma base de dados de 2.6 TB. A abordagem consome apenas 3,2% do tempo de processamento e 2% de novas instâncias em comparação com técnicas tradicionais.*

1. Introdução

Para lidar com ataques ou tentativas de intrusão de rede administradores recorrem a Sistemas de Detecção de Intrusão Baseada em Rede (NIDS) [Molina-Coronado et al. 2020], utilizando a estratégia de detecção baseada em comportamento. Neste cenário, as técnicas assumem que novos ataques serão detectados, mesmo que desconhecidos, já que o comportamento avaliado será substancialmente diferente de eventos benignos já conhecidos [Sommer and Paxson 2010].

Devido a vasta quantidade de dados coletados dos sensores de um sistema de detecção de intrusão, a literatura científica recorre a técnicas de Aprendizagem de Máquina (AM) para distinguir se um evento de rede é normal ou um ataque [Ahmad et al. 2021]. No entanto, como um modelo de AM é treinado com um conjunto de dados que representa um determinado período de coleta dos eventos, ele é suscetível a mudanças ao longo do tempo [Viegas et al. 2019]. Na prática, caso ele não seja atualizado com dados mais recentes com certa periodicidade, o sistema não será capaz de identificar corretamente novos ataques [Sommer and Paxson 2010].

Técnicas baseadas em AM requerem que o rótulo de cada evento esteja prontamente disponível. Isso torna indispensável a assistência humana na identificação de

ataques, uma vez que é comum que esta etapa seja realizada manualmente, demandando de semanas a meses para obter um resultado relevante [Blaise et al. 2020]. Além disso, o alto consumo de recursos computacionais torna a AM inaplicável em um contexto real [Molina-Coronado et al. 2020]. Isso se deve ao fato de que são necessários muitos eventos de rede para o treinamento, exigindo poder de processamento e armazenamento. Assim, é inviável que mão de obra especializada verifique milhões de eventos de rede em tempo suficiente para atualizar o modelo.

Na prática, a atualização de modelos de AM tradicionais ocorre através do descarte do modelo antigo, sendo necessário o retreinamento a partir de novos dados, o que implica em um alto custo de processamento e armazenamento. Como alternativa existe a AM baseada por fluxo, onde novos eventos são incorporados ao modelo de maneira incremental. No entanto, trabalhos que aplicam tal técnica consideram que todos os rótulos estarão sempre disponíveis [Din et al. 2020], o que é um cenário irreal, pois requer que muitos eventos de rede sejam rotulados a tempo.

Esta dissertação avança o estado da arte ao apresentar um novo esquema de detecção de intrusão baseado em aprendizado de fluxo, propondo atualizações de modelo, com a rejeição de instâncias antigas, sem afetar significativamente a acurácia. O esquema é implementado em três etapas: (1) Utiliza um conjunto de classificadores de aprendizado de fluxo para atualizações incrementais, reduzindo custos computacionais; (2) Avalia a qualidade da classificação com uma estratégia de opção de rejeição, aceitando apenas classificações com alta confiabilidade; e (3) Aplica atualizações incrementais ao modelo usando instâncias rejeitadas antigas, mantendo a acurácia ao longo do tempo. Assim, este trabalho contribui para a literatura ao reduzir custos computacionais de treinamento e armazenamento, mantendo altas taxas de acerto, mesmo com modelos desatualizados.

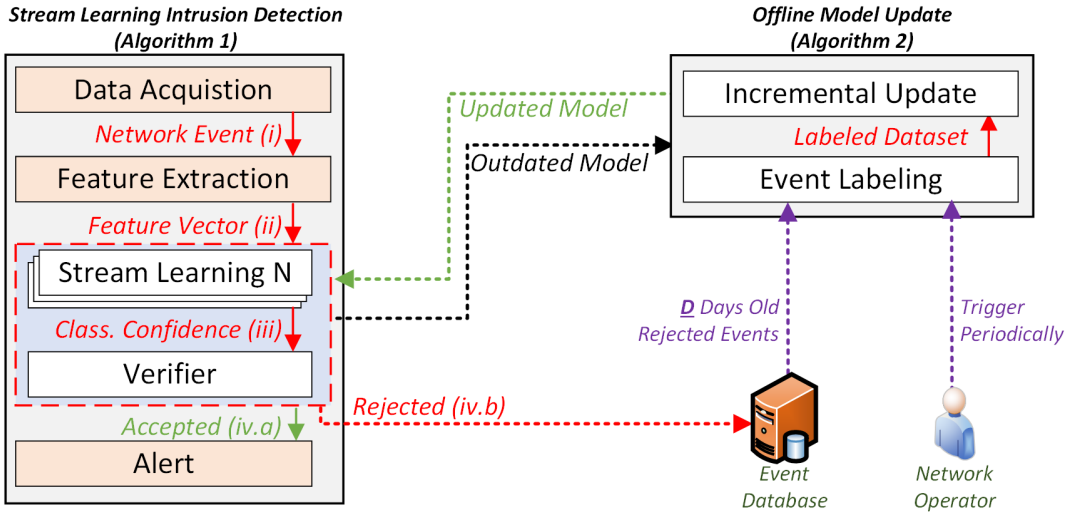
2. Atualizações Confiáveis

O trabalho de mestrado propôs uma abordagem que visa manter a acurácia do sistema ao longo do tempo e facilitar a atualização do modelo. Isso é feito através de dois processos: *Detecção de Intrusão com AM por Fluxo* e *Atualização de Modelo Offline*. O primeiro busca manter a acurácia do sistema mesmo sem atualizações. O segundo procura simplificar a atualização usando instâncias rejeitadas durante a produção. A Figura 1 ilustra ambos os procedimentos, onde as subseções 2.1 e 2.2 descrevem e detalham os procedimentos.

2.1. Detecção de Intrusão com AM por Fluxo

Os NIDS tradicionais efetuam a classificação para todas as instâncias e, devido a mudanças no tráfego de rede, aumenta substancialmente a taxa de erro ao longo do tempo. Para tratar isso, o trabalho de mestrado utiliza duas abordagens de classificação. Primeiro, para facilitar as atualizações do modelo, utiliza um conjunto de classificadores de aprendizado de fluxo. Isso permite atualizações incrementais do modelo, reduzindo os custos computacionais. Segundo, para manter a acurácia da classificação por períodos mais longos sem atualizações periódicas do modelo, utiliza classificação com opção de rejeição. Os eventos são classificados com base na confiança da classificação, aceitando apenas as classificações que ultrapassam um limiar pré-definido. A definição do limiar de rejeição é indispensável e depende da decisão do operador da rede. Uma taxa mais alta aumenta

Figura 1. Arquitetura para detecção de intrusão baseada em AM por fluxo com atualizações atrasadas.



a confiabilidade, porém mais eventos são rejeitados. Por outro lado, uma taxa mais baixa rejeita menos instâncias, mas pode levar a taxas de erro mais altas com o tempo. Eventos aceitos são encaminhados para o módulo de alerta (vide Figura 1, *iv.a*), enquanto eventos rejeitados são armazenados para futuras atualizações do modelo (vide Figura 1, *iv.b*). Ainda, no processo de classificação, a decisão final é determinada por votação majoritária.

Esta abordagem visa manter a acurácia do sistema ao longo do tempo e facilitar as atualizações do modelo, proporcionando uma resposta eficaz às mudanças no tráfego de rede.

2.2. Atualizações Offline

O trabalho propõe atualizações de modelo offline, mantendo o modelo desatualizado em produção. Isso reduz o número de eventos a serem rotulados ao longo do tempo, utilizando apenas os rejeitados previamente pelo sistema. As atualizações ocorrem após certo período, definido pelo operador da rede, quando os rótulos dos eventos são conhecidos.

O procedimento de atualização é acionado periodicamente, coletando e rotulando eventos rejeitados para atualizar os classificadores de aprendizado de fluxo. Essa abordagem simplifica a atualização do modelo, reduzindo o tempo de processamento e a necessidade de armazenamento, usando apenas eventos relevantes.

2.3. Discussão

O esquema proposto seleciona instâncias para atualização do modelo usando uma abordagem de rejeição na classificação, reduzindo o número de eventos de rede necessários para atualização. Essas instâncias são armazenadas temporariamente, permitindo fácil rotulação dos eventos recém coletados. Isso diminui os custos computacionais, pois apenas um subconjunto de instâncias é utilizado para atualizações, mantendo os modelos atualizados ao longo do tempo.

Como o modelo aceita apenas classificações com alta confiabilidade, as taxas de acurácia e a vida útil do modelo se mantém em produção, mesmo com modelos desatua-

lizados. Assim, nosso modelo enfrenta os principais desafios das atualizações em NIDS baseados em AM, garantindo confiabilidade na detecção de intrusões ao longo do tempo.

3. Avaliação Experimental

Esta seção descreverá os experimentos realizados. Dessa forma, espera-se que a proposta seja mais eficiente em termos de custo computacional, taxas de acerto e tempo de vida do modelo de AM. Na subseção 3.1 será descrito o *dataset*, enquanto que as subseções 3.2 e 3.3 detalham os experimentos.

3.1. Dataset

Diversos trabalhos carecem de *datasets* que simulem um ambiente de rede real. Neste trabalho, foi utilizado um dataset real do Samplepoint-F do MAWI [MAWI 2021], composto por tráfego diversificado e válido, coletado diariamente durante 15 minutos de uma conexão entre o Japão e os EUA. Para os experimentos, considerou-se o ano de 2014, com eventos agrupados em intervalos de 15 segundos e 66 características extraídas pela abordagem de Moore [Moore and Zuev 2005].

Devido ao desbalanceamento e para evitar viés entre as classes *normal* e *ataque*, foi feita uma subamostragem aleatória da classe majoritária nos dados de treinamento, sem repetição, mantendo a proporção entre as classes.

3.2. Construção do Modelo

Um esquema para linha de base foi implementado utilizando os classificadores de fluxo Hoeffding Tree (HT), Leveraging Bag (Bag) e Oza Bagging (Oza). O modelo proposto no trabalho depende de um conjunto de classificadores incluindo os modelos citados, semelhante à abordagem de *Ensemble*. Os classificadores foram implementados usando a API `scikit-multiflow` v.0.5.3 e os mesmos parâmetros do caso anterior foram usados.

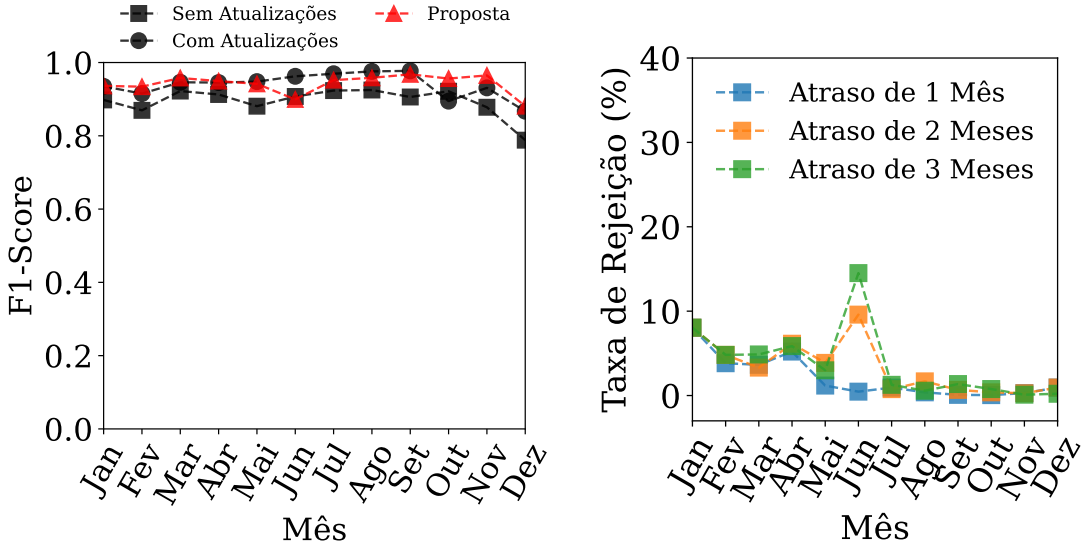
O classificador HT foi avaliado com critério de divisão de nó de ganho de informação, um *grace period* de 200 e previsão de nó folha de naive Bayes adaptativo. O Bag foi avaliado com 3 HT como classificador base e ADWIN como algoritmo de alavancagem. O Oza também foi avaliado com 3 HT como base. Um *ensemble* foi formado através de votação majoritária dos classificadores HT, Bag e Oza.

Por fim, é importante ressaltar que as atualizações periódicas consideram uma janela de 30 dias para atualização, ou seja, no primeiro dia de fevereiro será realizada a atualização com o primeiro dia de janeiro.

3.3. Tratando Mudanças no Comportamento da Rede

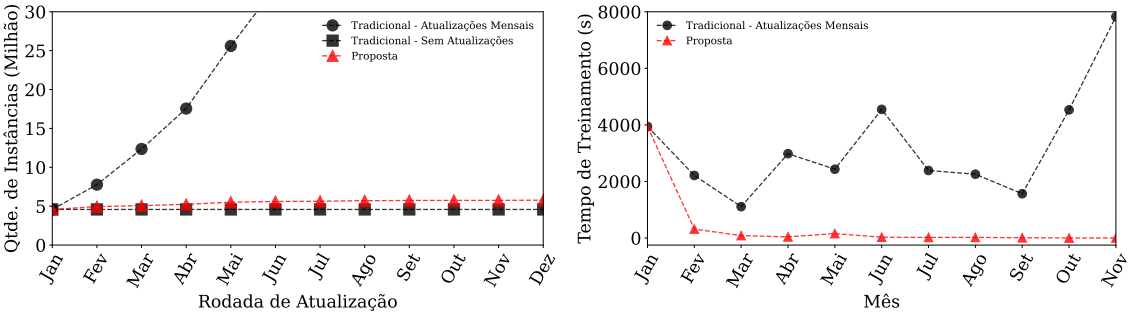
A acurácia do sistema é prejudicada com a alta variabilidade do tráfego da rede. Isso necessita que sejam realizadas atualizações periódicas para evitar vulnerabilidades no NIDS. Sistemas baseados em técnicas de AM de aprendizagem em lotes precisam reconstruir o modelo do zero para atualizar. Por outro lado, o uso de técnicas de AM de aprendizagem por fluxo elimina essa necessidade, reduzindo os custos computacionais em termos de tempo de processamento e armazenamento de instâncias para atualização.

Figura 2. Comparação do F1-Score da abordagem proposta contra as demais e a taxa de rejeição ao longo do tempo.



(a) F1-Score (b) Taxa de Rejeição

Figura 3. Comparação do custo computacional (processamento e armazenamento) ao longo do tempo.



(a) Quantidade de Instâncias (b) Tempo de Treinamento

Por exemplo, a Figura 2(a) ilustra o F1-Score ao longo dos meses do ano, para três abordagens distintas: Modelos **sem** atualização, modelos **com** atualizações e a abordagem proposta. Além disso, é notável como a proposta ainda se sobressai em relação à abordagem de aprendizagem por fluxo, onde espera-se um resultado mais satisfatório por não ter de reconstruir um modelo do zero.

Para a tarefa de atualização, são utilizadas instâncias que foram previamente rejeitadas de acordo com uma taxa de aceitação de 10% de erro. A Figura 2(b) mostra a relação da taxa de rejeição ao longo do tempo, com diferentes períodos de atraso. É notável que a relação entre tempo e quantidade de instâncias rejeitadas é inversamente proporcional, pois são necessárias menos instâncias para que sejam realizadas atualizações.

Por fim, a Figura 3 compara o custo computacional da proposta em termos de armazenamento de instâncias e tempo de treinamento do modelo. Apesar da melhoria na acurácia do modelo, foi reduzido a complexidade da atualização do modelo. O esquema

Tabela 1. Resumo das produções científicas do autor.

Referência	Tipo	Qualis	Local	Citações
[Ramos et al. 2021]	Conferência	A1	IEEE ICC	26
[de Oliveira et al. 2024]	Conferência	A1	IEEE IJCNN	–
[Horchulhack et al. 2022b]	Conferência	A1	IEEE GLOBECOM	2
[Horchulhack et al. 2024a]	Periódico	A1	JNCA	2
[Horchulhack et al. 2022d]	Conferência	A1	GLOBECOM	1
[Horchulhack et al. 2022c]	Periódico	A1	Computer Networks	24
[de Oliveira et al. 2023]	Conferência	A2	IEEE Trustcom	–
[Horchulhack et al. 2024c]	Conferência	A2	IEEE IWCMC	–
[Geremias et al. 2023]	Conferência	A2	IEEE CCNC	2
[Geremias et al. 2022]	Conferência	A2	IEEE IWCMC	14
[Oliveira et al. 2024]	Conferência	A3	AINA	–
[Horchulhack et al. 2023]	Conferência	A4	SBSeg	–
[Horchulhack et al. 2024b]	Conferência	A4	SBRC	–
[Horchulhack et al. 2022a]	Conferência	B1	CSNet	3

Tabela 2. Resumo dos softwares registrados constando o autor.

Número de Registro	Título
512024000703-2	Um software para viabilizar atualizações de modelos de aprendizagem de máquina para detecção de intrusões baseada em rede
512024000711-3	Kubemon: extrator de métricas de desempenho de sistema operacional e aplicações conteinerizadas em ambientes de nuvem no domínio do provedor
512024000710-5	Uma ferramenta para detecção hierárquica e confiável de malwares Android por meio de CNN baseada em imagens

proposto aumentou a acurácia, exigindo poucas instâncias rotuladas em comparação com as técnicas tradicionais de atualização mensal.

4. Conclusão e Impacto da Dissertação

As Tabelas 1 e 2 apresentam os trabalhos publicados e os softwares registrados durante o mestrado. Como autor principal, foram produzidas 5 conferências, 2 periódicos e 2 registros de software, além de participação como coautor em outras 6 conferências e 1 registro de software. Os trabalhos relacionados à dissertação foram desenvolvidos através de uma iniciativa de PIBIC Master (*Double Degree*), premiada como melhor projeto de iniciação científica das ciências exatas de 2021 da universidade. A repercussão é evidente pela quantidade de produções e citações.

A dissertação encontra-se disponível no link `https://www.ppgia.pucpr.br/pt/arquivos/mestrado/dissertacoes/2023/Atualizacao_Confiavel_Modelos_Deteccao_Intrusao_Baseada_Aprendizagem_Maquina.pdf`

Agradecimentos

Este trabalho foi parcialmente financiado pelo Conselho Nacional de Desenvolvimento Científico e Tecnológico (CNPq), processos nº 304990/2021-3, 407879/2023-4 e 302937/2023-4.

Referências

[Ahmad et al. 2021] Ahmad, Z., Shahid Khan, A., Wai Shiang, C., Abdullah, J., and Ahmad, F. (2021). Network intrusion detection system: A systematic study of machine learning and deep learning approaches. *Transactions on Emerging Telecommunications Technologies*, 32(1):e4150.

[Blaise et al. 2020] Blaise, A., Bouet, M., Conan, V., and Secci, S. (2020). Detection of zero-day attacks: An unsupervised port-based approach. *Computer Networks*, 180:107391.

[de Oliveira et al. 2023] de Oliveira, P. R., Santin, A. O., Horchulhack, P., and Viegas, E. K. (2023). A dynamic network-based intrusion detection model for industrial control systems. In *IEEE International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*.

[de Oliveira et al. 2024] de Oliveira, P. R., Viega, E., Santin, A., Horchulhack, P., and de Matos, E. (2024). Toward a reliable network-based intrusion detection model for scada: A classification with reject option approach. In *International Joint Conference on Neural Networks (IJCNN)*.

[Din et al. 2020] Din, S. U., Shao, J., Kumar, J., Ali, W., Liu, J., and Ye, Y. (2020). Online reliable semi-supervised learning on evolving data streams. 525:153–171.

[Geremias et al. 2022] Geremias, J., Viegas, E. K., Santin, A. O., Britto, A., and Horchulhack, P. (2022). Towards multi-view android malware detection through image-based deep learning. In *International Wireless Communications and Mobile Computing (IWCMC)*, pages 572–577.

[Geremias et al. 2023] Geremias, J., Viegas, E. K., Santin, A. O., Britto, A., and Horchulhack, P. (2023). Towards a reliable hierarchical android malware detection through image-based cnn. In *2023 IEEE 20th Consumer Communications & Networking Conference (CCNC)*, pages 242–247. IEEE.

[Horchulhack et al. 2023] Horchulhack, P., Viegas, E., Santin, A., and Ramos, F. (2023). Kubemon: extrator de métricas de desempenho de sistema operacional e aplicações conteinerizadas em ambientes de nuvem no domínio do provedor. In *Anais Estendidos do XXIII Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*, pages 25–32.

[Horchulhack et al. 2022a] Horchulhack, P., Viegas, E. K., and Lopez, M. A. (2022a). A stream learning intrusion detection system for concept drifting network traffic. In *2022 6th Cyber Security in Networking Conference (CSNet)*, pages 1–7. IEEE.

[Horchulhack et al. 2022b] Horchulhack, P., Viegas, E. K., and Santin, A. O. (2022b). Detection of service provider hardware over-commitment in container orchestration environments. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. IEEE.

[Horchulhack et al. 2022c] Horchulhack, P., Viegas, E. K., and Santin, A. O. (2022c). Toward feasible machine learning model updates in network-based intrusion detection. *Computer Networks*, 202:108618.

[Horchulhack et al. 2022d] Horchulhack, P., Viegas, E. K., Santin, A. O., and Geremias, J. (2022d). Intrusion detection model updates through gan data augmentation and transfer learning. In *GLOBECOM 2022 - 2022 IEEE Global Communications Conference*. IEEE.

[Horchulhack et al. 2024a] Horchulhack, P., Viegas, E. K., Santin, A. O., Ramos, F. V., and Tedeschi, P. (2024a). Detection of quality of service degradation on multi-tenant containerized services. *Journal of Network and Computer Applications*, 224:103839.

[Horchulhack et al. 2024b] Horchulhack, P., Viegas, E. K., Santin, A. O., and Simioni, J. A. (2024b). Fortalecendo a segurança de redes: Um olhar profundo na detecção de intrusões com cnn baseada em imagens e aprendizado por transferência. In *SBRC 2024 - XLII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*.

[Horchulhack et al. 2024c] Horchulhack, P., Viegas, E. K., Santin, A. O., and Simioni, J. A. (2024c). Network-based intrusion detection through image-based cnn and transfer learning. In *International Wireless Communications & Mobile Computing Conference (IWCMC)*.

[MAWI 2021] MAWI (2021). MAWI Working Group Traffic Archive - Samplepoint F.

[Molina-Coronado et al. 2020] Molina-Coronado, B., Mori, U., Mendiburu, A., and Miguel-Alonso, J. (2020). Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. *IEEE Trans. on Network and Service Management*, 17(4):2451–2479.

[Moore and Zuev 2005] Moore, A. W. and Zuev, D. (2005). Internet traffic classification using bayesian analysis techniques. In *Proceedings of the 2005 ACM SIGMETRICS international conference on Measurement and modeling of computer systems - SIGMETRICS '05*. ACM Press.

[Oliveira et al. 2024] Oliveira, J., Santin, A., Viega, E., and Horchulhack, P. (2024). A non-interactive one-time password-based method to enhance the vault security. In *The 38-th International Conference on Advanced Information Networking and Applications (AINA)*.

[Ramos et al. 2021] Ramos, F., Viegas, E., Santin, A., Horchulhack, P., dos Santos, R. R., and Espindola, A. (2021). A machine learning model for detection of docker-based app overbooking on kubernetes. In *IEEE International Conference on Communications*, pages 1–6.

[Sommer and Paxson 2010] Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*. IEEE.

[Viegas et al. 2019] Viegas, E., Santin, A., Bessani, A., and Neves, N. (2019). BigFlow: Real-time and reliable anomaly-based intrusion detection for high-speed networks. *Future Generation Computer Systems*, 93:473–485.