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 instances needed for updates and computational costs. Instances rejected in classification are stored for incremental update, allowing automatic labeling from public repositories. Experiments with a 2.6TB database showed that the proposal maintains high accuracy rates 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%, while 8% of events are rejected. This approach consumes 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 intrusões 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

De acordo com a Kaspersky, aproximadamente 600 milhões de ataques de rede foram bloqueados por soluções de segurança apenas em 2020[Kaspersky 2020]. Para lidar com a crescente onda de ataques à rede, os administradores tipicamente recorrem aSistemas 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[Gates and Taylor 2006], já que o comportamento avaliado será substancialmente diferente de eventos benignos já conhecidos [Sommer and Paxson 2010].

Devido à 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].

Apesar das técnicas baseadas em AM serem promissoras na identificação de ataques de rede, elas geralmente 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 e, consequentemente, é necessário grande capacidade de 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 [Molina-Coronado et al. 2020], 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.

Neste cenário, a presente dissertação avança o estado da arte ao apresentar um novo esquema de detecção de intrusão baseado em aprendizado de fluxo. O trabalho propõe atualizações de modelo com instâncias antigas rejeitadas sem afetar a acurácia da classificação. O esquema é implementado em três etapas. Primeiramente, o modelo utiliza um conjunto de classificadores de aprendizado de fluxo para realizar atualizações incrementais do modelo, reduzindo os custos computacionais. Em seguida, a abordagem avalia a qualidade da classificação através da estratégia de opção de rejeição, aceitando apenas classificações altamente confiáveis, mesmo com modelos desatualizados. Por fim, as atualizações do modelo são aplicadas de forma incremental utilizando instâncias rejeitadas antigas, facilitando a atualização do modelo e garantindo sua acurácia ao longo do tempo. Desta maneira, o presente trabalho contribui para a literatura científica ao apresentar uma estratégia para reduzir custos computacionais sobre o treinamento e armazenamento de instâncias, mantendo altas taxas de acerto ao longo do tempo, mesmo que o modelo subjacente esteja desatualizado.

2. Trabalhos Relacionados

Esta seção apresenta um resumo da revisão de literatura sobre o tema do trabalho. Ao buscar uma solução de detecção de intrusão baseada em rede que se adapte à evolução do tráfego, é necessário considerar três aspectos gerais. O primeiro é um cenário de tráfego de rede altamente dinâmico. O segundo é o custo computacional, representado pelo processamento e armazenamento resultante das atualizações. Por fim, é necessário um esquema de classificação confiável capaz de melhorar as acurácias do sistema.

Desta maneira, em relação aos trabalhos encontrados na literatura e em comparação ao em questão, são listados cinco métricas, sendo: Ensemble de Classificadores (EC), Aprendizagem por Fluxo (AF), Atualizações Periódicas (AP), Avaliação de Classificação (AC), Avaliação de Custo Computacional (ACC). Neste cenário, dos estudos elenca-

Tabela 1. Resumo das características dos trabalhos relacionados

Trabalho	Características				
	Ensemble de Classificadores	**Aprendizagem por Fluxo**	**Atualizações Periódicas**	**Avaliação de Classificação**	**Avaliação de Custo Computacional**
[Viegas et al. 2017]			✓	✓	
[Adhikari et al. 2018]		✓	✓		
[Martindale et al. 2020]	✓	✓	✓		
[Li et al. 2020]			✓		✓
[Viegas et al. 2020]		✓	✓	✓	
[Das et al. 2022]	✓				

dos na Tabela 2, todas as técnicas consideram AM como principal mecanismo de classificação. Além disso, foram selecionados os trabalhos que possuem, no mínimo, duas das características específicas citadas.

Os trabalhos anteriores focam em melhorar a acurácia dos NIDS, mas é notável que essas melhorias raramente são implementadas em ambientes reais devido à alta variação no comportamento do tráfego de rede, o que prejudica os classificadores a longo prazo. É importante ressaltar que os classificadores precisam de atualizações periódicas para se adaptarem a mudanças, porém muitos estudos não consideram o alto impacto do consumo de recursos computacionais para implementar, eficientemente, em cenários realistas. Além disso, é pontuado a importância da confiabilidade das classificações, sugerindo que abordagens para melhorar tal característica pode reduzir os falsos-positivos e falsos-negativos, aumentando a confiabilidade do sistema. Assim, este trabalho procura preencher tais lacunas.

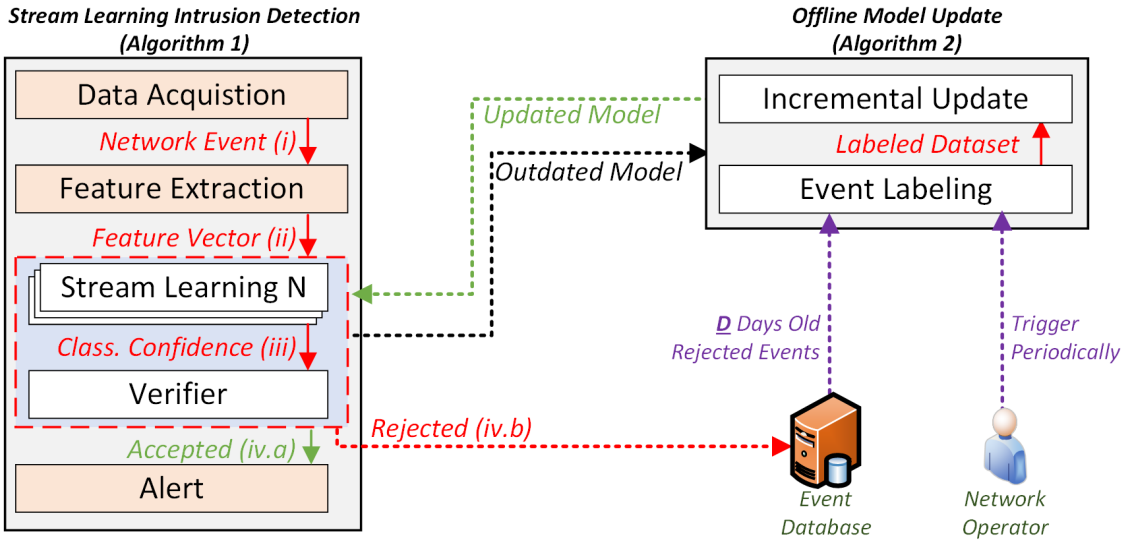
3. Atualizações Confiáveis

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 3.1 e 3.2 descrevem e detalham os procedimentos.

3.1. Detecção de Intrusão com AM por Fluxo

Os NIDS tradicionais efetuam a classificação para todas as instâncias, aumentando substancialmente a taxa de erro ao longo do tempo devido a mudanças no tráfego de rede. Para tratar isso, o trabalho de mestrado utiliza duas abordagens de classificação. Primeiro, para facilitar as atualizações do modelo, o trabalho 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

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



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 crucial e depende da decisão do operador da rede. Uma taxa mais alta aumenta 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.

3.2. Atualizações Offline

O trabalho propõe realizar atualizações de modelo offline, mesmo mantendo o modelo desatualizado em uso no ambiente de produção. Isso é feito para reduzir o número de eventos que precisam ser rotulados ao longo do tempo, utilizando apenas aqueles que foram rejeitados anteriormente pelo sistema. As atualizações do modelo são conduzidas apenas após um certo período desde a rejeição de um evento, garantindo que os eventos mais antigos possam ser rotulados autonomamente. Isso é possível porque, até o momento da atualização do modelo, os rótulos adequados dos eventos serão publicamente conhecidos.

O procedimento de atualização do modelo é acionado periodicamente e envolve a coleta de eventos rejeitados. Sua rotulagem e a aplicação desses rótulos são utilizados para atualizar incrementalmente os classificadores de aprendizado de fluxo em uso no ambiente de produção. Essa abordagem simplifica significativamente a tarefa de atualização do modelo, reduzindo o tempo de processamento e a necessidade de armazenamento, enquanto garante que apenas eventos relevantes sejam usados para as atualizações.

3.3. Discussão

O número de eventos de rede necessários para atualização diminui à medida que o esquema proposto é capaz de selecionar quais instâncias devem ser usadas para atualização do modelo por meio de uma abordagem de rejeição na classificação. Além disso, como essas instâncias são armazenadas por um período de tempo antes de serem utilizadas para atualizações do modelo, os eventos de rede recém-coletados podem ser facilmente rotulados, já que são armazenados por um tempo antes de serem usados para atualizações do modelo. Isso reduz os custos computacionais, uma vez que apenas um subconjunto de instâncias é utilizado para atualizações do modelo, e os modelos desatualizados são atualizados ao longo do tempo.

Finalmente, considerando que o modelo aceita apenas classificações com alta confiabilidade, as taxas de acurácia e a vida útil do modelo são melhoradas durante o uso em produção, mesmo com modelos desatualizados, já que apenas classificações altamente confiantes são aceitas pelo nosso esquema. Como resultado, nosso modelo proposto pode enfrentar os principais desafios relacionados às atualizações de modelo em NIDS baseados em AM, proporcionando confiabilidade na detecção de intrusões ao longo do tempo.

4. 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 deAM. Na subseção 4.1 será descrito o *dataset*, enquanto que as subseções 4.2 e 4.3 detalha os experimentos.

4.1. Dataset

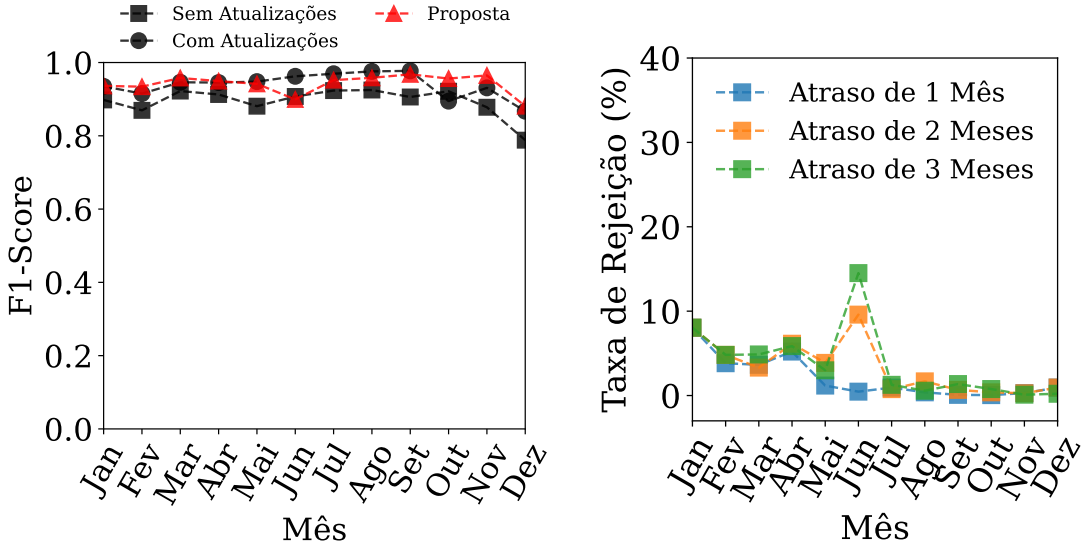
Diversos trabalhos do estado-da-arte carecem de *datasets* que simulem um comportamento de um ambiente de rede real. Nesse trabalho foi considerado um dataset real, consistindo em fluxos de rede que passam através do Samplepoint-F do arquivo MAWI [MAWI 2021], sendo composta por tráfego real, válido e altamente diverso. A coleta é realizada diariamente durante 15 minutos, de uma interconexão entre o Japão e os Estados Unidos da América. Para os experimentos do trabalho foi considerado o ano de 2014, onde agrupa os eventos de em intervalos de 15 segundos. Ao final, são extraídas 66 características de fluxos da abordagem de Moore [Moore and Zuev 2005].

Como a base apresenta um cenário real, a distribuição das classes *normal* e *ataque* está desbalanceada, que poderá prejudicar o processo de treinamento do modelo. Para solucionar o problema foi feita uma subamostragem aleatória dos dados de treinamento da classe majoritária, sem repetição, mantendo equivalente a proporção entre as classes.

4.2. Construção do Modelo

Um esquema proposto foi implementado utilizando os classificadores de fluxo Hoeffding Tree (HT), Leveraging Bag (Bag) e Oza Bagging (Oza). O modelo proposto 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 foram usados. Devido ao desequilíbrio do conjunto de dados, uma técnica de subamostragem aleatória sem reposição é aplicada como procedimento de pré-processamento.

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

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.

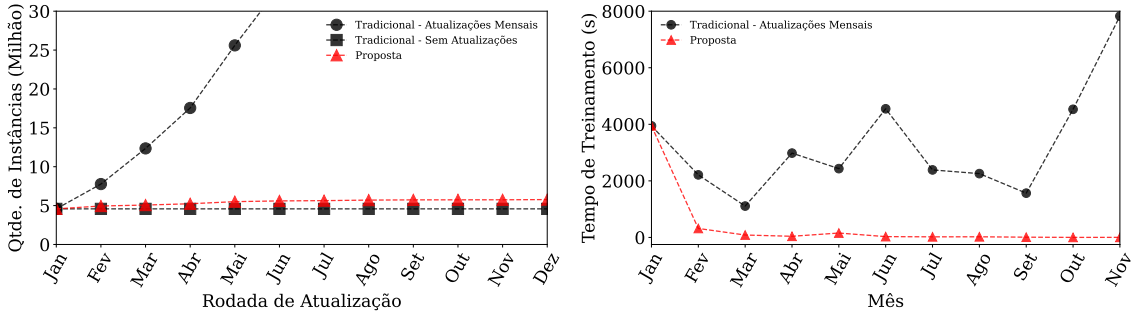
4.3. Tratando Mudanças no Comportamento da Rede

Quando existe variabilidade no tráfego da rede, isto é, existe uma mudança do comportamento previamente observada, prejudica a acurácia do sistema. Dessa forma, são necessárias que atualizações periódicas sejam feitas para que o sistema não fique vulnerável. Sistemas de detecção de intrusão que são baseados em técnicas de AM de aprendizagem em lotes precisam que o modelo seja reconstruído do zero para corresponder a uma atualização. Por outro lado, se empregar técnicas de AM de aprendizagem por fluxo, não será necessária essa reconstrução, diminuindo custos computacionais em relação ao tempo de processamento e ao armazenamento de instâncias para atualização.

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

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



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

Tabela 2. 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	–
[Horchulhack et al. 2022d]	Conferência	A1	GLOBECOM	1
[Horchulhack et al. 2022c]	Periódico	A1	Computer Networks	22
[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	11
[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	1

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, reduzimos significativamente a complexidade da atualização do modelo. Nosso esquema proposto aumentou a acurácia, exigindo apenas uma quantidade mínima das instâncias rotuladas necessárias em comparação com as técnicas tradicionais de atualização mensal. Além disso, o custo computacional de nossa abordagem é menor, representando apenas uma pequena parcela dos custos da abordagem tradicional de conjunto com atualizações mensais de modelo, conforme ilustrado na Figura 3(b).

5. Conclusão e Impacto da Dissertação

As Tabelas 2 e 3 apresentam os trabalhos publicados e os softwares registrados durante o período do mestrado, respectivamente. Através da quantidade de produções, bem como o número de citações, é evidente a repercussão causada durante o trabalho. Em termos

Tabela 3. 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

práticos, como autor principal foram produzidas 5 conferências, 2 periódicos e 3 registros de software. Além disso, o autor participou como coautor na produção de outras 6 conferências. Os trabalhos relacionados à dissertação foram desenvolvidos através de uma iniciativa de PIBIC Master (*Double Degree*), sendo inclusive agraciado com o prêmio de melhor projeto de iniciação científica de 2021 da universidade, período em que o autor iniciou os estudos no mestrado. Portanto, o impacto da dissertação de mestrado torna-se notável considerando os resultados obtidos durante sua realização.

Agradecimentos

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

Referências

[Adhikari et al. 2018] Adhikari, U., Morris, T. H., and Pan, S. (2018). Applying hoeffding adaptive trees for real-time cyber-power event and intrusion classification. *IEEE Transactions on Smart Grid*, 9(5):4049–4060.

[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.

[Das et al. 2022] Das, S., Saha, S., Priyoti, A. T., Roy, E. K., Sheldon, F. T., Haque, A., and Shiva, S. (2022). Network intrusion detection and comparative analysis using ensemble machine learning and feature selection. *IEEE Transactions on Network and Service Management*, 19(4):4821–4833.

[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.

[Gates and Taylor 2006] Gates, C. and Taylor, C. (2006). Challenging the anomaly detection paradigm: A provocative discussion. In *Proc. of the Workshop on New Security Paradigms (NSPW)*, pages 21–29.

[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)*.

[Kaspersky 2020] Kaspersky (2020). *Kaspersky Security Bulletin 2020. Statistics*.

[Li et al. 2020] Li, X., Chen, W., Zhang, Q., and Wu, L. (2020). Building auto-encoder intrusion detection system based on random forest feature selection. *Computers & Security*, 95:101851.

[Martindale et al. 2020] Martindale, N., Ismail, M., and Talbert, D. A. (2020). Ensemble-based online machine learning algorithms for network intrusion detection systems using streaming data. *Information*, 11(6):315.

[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.

[Viegas et al. 2020] Viegas, E. K., Santin, A. O., Cogo, V. V., and Abreu, V. (2020). A reliable semi-supervised intrusion detection model: One year of network traffic anomalies. In *ICC 2020 - 2020 IEEE International Conference on Communications (ICC)*. IEEE.

[Viegas et al. 2017] Viegas, E. K., Santin, A. O., and Oliveira, L. S. (2017). Toward a reliable anomaly-based intrusion detection in real-world environments. *Computer Networks*, 127:200–216.