Avaliação da capacidade de generalização de IDS *stateful* utilizando aprendizado de máquina

Marcelo Fernandes Domingues[1,2], Gustavo de C. Bertoli[1], Leonardo H. de Melo[1], Osamu Saotome[1], Aldri Santos[3], Lourenço Alves Pereira[1]

[1]Pós-Graduação em Engenharia Eletrônica e Computação (PG/EEC)
Instituto Tecnológico de Aeronáutica (ITA) – São José dos Campos, SP – Brazil

[2]Diretoria de Aeronáutica da Marinha (DAerM)
Marinha do Brasil – Rio de Janeiro, RJ – Brazil

[3]Departamento de Ciência da Computação
Universidade Federal de Minas Gerais (UFMG) – Belo Horizonte, MG – Brazil

{marcelo,bertoli,leonardo,osaotome,ljr}@ita.br, aldri@dcc.ufmg.br

Abstract. *Machine learning is relevant for the characterization of attacks on computer networks, as it allows the identification of traffic patterns and, therefore, the implementation of mechanisms for blocking malicious actions. However, the capacity of solutions to generalize in exogenous contexts is still missing. Hence, we evaluated the performance of different models, sush as DT, LR, MLP, NB, SVM, and XGB, in the UNSW-NB15, CICIDS-2017, BoT-IoT, ToN-IoT, and AB-TRAP datasets. As a result, we observed low levels of generalization in the models. Furthermore, feature engineering enables the comparison of models and leverages the learning process. Finally, we analyze the effectiveness of attributes as predictors of scanning.*

Resumo. *Aprendizado de máquina é relevante para a caracterização de ataques em redes de computadores, permitindo identificar padrões de tráfego e, com isso, implementar mecanismos que bloqueiam ações maliciosas. No entanto, pouco se discute sobre a capacidade de generalização das soluções em diferentes contextos operacionais. Este trabalho avalia o desempenho de diferentes modelos, como DT, LR, MLP, NB, SVM e XGB, nos conjuntos de dados UNSW-NB15, CICIDS-2017, BoT-IoT, ToN-IoT e AB-TRAP. Como resultado, observou-se uma baixa capacidade de generalização. Mais ainda, a engenharia de atributos facilitou a comparação entre os modelos e contribuiu no processo de aprendizagem. Por fim, analisou-se a efetividade de atributos como preditores de scanning em redes.*

1. Introdução

Nossa sociedade passa por uma grande revolução habilitada pela ubiquidade da Internet e digitalização de serviços que operam nessa infraestrutura. Assim, serviços de comércio online e *streaming* tornam-se mais comuns em nosso dia-a-dia, fato complementado pelo aumento abrupto do trabalho remoto como medida de distanciamento social em resposta a pandemia do COVID-19. Dessa forma, a condição atual já reflete um grande volume de tráfego de dados e é escalada por novas perspectivas

tecnológicas representadas por uma maior adoção de dispositivos de Internet das Coisas (IoT) e 5G. Portanto, percebe-se uma crescente diversidade de aplicações e perfis de tráfego, pois IoT e 5G servem de habilitadores para acomodar classes de serviços hoje ainda impraticados [Al-Sarawi et al. 2020].

A segurança cibernética é um requisito não funcional e proporciona condições necessárias para definição confiável de serviços e aplicações cuja infraestrutura de comunicação baseia-se na Internet. Ainda, serviços críticos beneficiam-se cada vez mais dos recursos proporcionados pela Internet. Por outro lado, é evidente que disrupções nesses sistemas podem levar a cenários catastróficos. Assim, uma das soluções amplamente utilizadas para a segurança de redes são os sistemas de detecção de intrusão (*Intrusion Detection System* - IDS). Esse tipo de solução, que inicialmente trabalhava com assinaturas conhecidas de ataques, atualmente, encontra-se em constante evolução a partir da introdução de técnicas de aprendizado de máquina (AM) e aprendizado profundo para a criação de uma nova geração de mecanismos de detecção de ações maliciosas. Consequentemente, as principais técnicas recentes para detecção de ataques em redes baseiam-se em aprendizado de máquina e mineração de dados, em que se destacam: redes neurais, associação de regras, redes bayesianas, *clustering* e árvores de decisão [Buczak and Guven 2016].

A área de pesquisa de IDSs baseados em AM é dependente de conjuntos de dados que permitem avaliar comparativamente técnicas e abordagens a fim de obter soluções mais eficientes [Kenyon et al. 2020]. Assim, avaliar a capacidade de generalização dos modelos de AM possibilita uma visão mais ampla sobre a qualidade dos dados presentes e a capacidade de modelos em identificar quais atributos são relevantes entre tráfegos com diferentes características. Especificamente, as atividades de *scanning* merecem um destaque pois geralmente pertencem às primeiras atividades realizadas por atacantes; logo, identificar, descartar e gerar alertas sobre essas ações é crucial para inviabilizar ataques [de Carvalho Bertoli et al. 2021b].

Este artigo demonstra a capacidade de generalização de seis modelos de aprendizado de máquina em cinco conjunto de dados. Os modelos árvore de decisão (*desicion tree*–DT), regressão logística (*logistic regression*–LR), *multi-layer perceptron*–MLP, *naïve bayes*–NB, *support vector machine*–SVM e *eXtreme Gradient Boost*–XGB, foram testados nos conjuntos de dados UNBW-NB15, CIC-IDS, ToN-IoT, BoT-IoT e AB-TRAP. CICFlowMeter, adotado neste trabalho, gera um conjunto de atributos que caracterizam de forma agregada todo o tráfego de um fluxo (stateful). Ele foi adotado para uniformizar o conjunto de preditores (atributos) utilizado pelos modelos. Com isso, observou-se a viabilidade na comparação de referência cruzada entre modelos e conjunto de dados . Por fim, discuti-se sobre os atributos de redes relevantes para a detecção de ataques de *scanning* em redes e a dificuldade de generalização existente neste tipo de problema.

O restante do artigo está organizado da seguinte maneira: na Seção 2 são discutidos os trabalhos relacionados. A Seção 3 descreve a proposta adotada neste artigo. A Seção 4 reporta os resultados obtidos em nossos experimentos comparando-os com trabalhos previamente publicados e com uma discussão sobre esses resultados. Na Seção 5, conclui-se este estudo e listamos trabalhos futuros.

2. Trabalhos Relacionados

A tendência de abordagem ao problema de classificação de tráfego malicioso e benigno ocorre geralmente por meio da escolha de um conjunto de dados conhecido pela comunidade científica e, a partir dele, treinam-se modelos capazes de identificar padrões de ações maliciosas. No entanto, pouco se discute sobre as consequências dessa abordagem. Portanto, é necessário verificar a independência dos modelos em relação ao conjunto de dados (*datasets*) adotado.

[Catillo et al. 2021] utiliza-se do tráfego de ataques de negação de serviço (*denial of service* - DoS) de cinco conjuntos de dados. Os ataques disponíveis nesses conjunto de dados são avaliados contra diferentes formas de defesa. Constatou-se que modelos treinados a partir desses dados não foram capazes de detectar ataques similares presentes em outros conjuntos. Os autores concluem que a generalização do aprendizado não foi obtida na prática e defendem a geração de conjuntos de dados de forma mais rigorosa. Muito embora nosso trabalho tenha semelhança, nosso enfoque, por outro lado, se dá para os ataques de *scanning*. Assim, utilizamos conjuntos de dados mais referenciados pela comunidade acadêmica e também introduzimos uma forma de uniformização de atributos entre os conjuntos de dados.

A dificuldade de transpor IDS (*Intrusion Detection System*) baseados em aprendizado de máquina (AM) ao contexto real de operação é apontado por [de Carvalho Bertoli et al. 2021a]. Neste trabalho, os autores apontam o envelhecimento dos conjuntos de dados com relação as características e evoluções tanto do tráfego normal quanto dos ataques de redes. Apontam o uso do framework AB-TRAP para geração de conjunto de dados mais atuais e a possibilidade de criar atributos *stateful* a partir do framework proposto. Apontam também para a necessidade de mudança de paradigma na pesquisa de IDS do contexto centrado em modelos (*model-centric*) para um contexto centrado na qualidade dos dados (*data-centric*) [de Carvalho Bertoli et al. 2021a]. Diferentemente, no presente artigo adotam-se atributos de fluxo (*stateful*), tráfego recente para o caso de ataques de *scanning* do próprio AB-TRAP e mantém-se o foco no teste da generalização do IDS baseado em AM para diversos contextos.

Um estudo comparativo entre o desempenho de modelos em múltiplos contextos, através de diferentes conjuntos de dados, é apresentado em [Layeghy and Portmann 2022]. Os autores prosseguem na tentativa de explicar os resultados com o uso do SHAP (*Shapley Additive Explanations*). No entanto, parte do resultado pode ser atribuído às distribuições distintas de tipos de ataques em cada conjunto de dados, uma vez que foi utilizada uma classe agregada para detecção binária. Além disso, somente modelos não supervisionados foram treinados com conjuntos de dados balanceados, o que, quando em uso somente da métrica F1-score, não evidencia o adequado desempenho dos modelos supervisionados. Anteriormente, [Al-Riyami et al. 2018] já havia comparado treinamento e teste em diferentes contextos, porém somente com o os conjuntos de dados NSL-KDD e gureKDD.

Para explorar os conjuntos de dados existentes, [Apruzzese et al. 2022] propuseram que fluxos de cada classe existente em cada conjunto de dados fossem combinados para criar, com conjuntos de treinamento e de teste, contextos diversos dos disponíveis em cada conjunto de dados individualmente. No entanto, o uso de flu-

xos benignos do mesmo conjunto de dados durante o treinamento e testes, conforme proposto, não reflete a capacidade de generalização necessária em aplicações reais, e o trabalho não explorou as características de ataques específicos separadamente ao utilizar três classes agregadas de ataques.

Uma grande dificuldade é a comparação de resultados obtidos por modelos treinados em diferentes conjuntos de dados. Normalmente, esses conjuntos de dados utilizam-se de conjuntos próprios de atributos que, assim, impossibilitam a aplicação direta de um modelo de AM já treinado. Estes atributos específicos a cada conjunto de dados, que advém da *expertise* de seus autores, são resultado da falta de um conjunto padrão conforme argumentado por [Sarhan et al. 2021]. Sarhan et al. argumentam que um conjunto padrão de atributos reduziria a complexidade da extração das características da rede em questão, e permitiria a avaliação da generalização dos modelos de AM em diversos contextos através da aplicação em diversos conjuntos de dados. No trabalho, os autores utilizam um conjunto de atributos `NetFlow`, produzindo dois conjuntos de atributos totalizando 12 e 43, respectivamente. Em seguida, comparam o desempenho entre os novos atributos propostos e os atributos originais em quatro conjuntos de dados comumente utilizados (UNSW-NB15, CIC-IDS2018, ToN-IoT e BoT-IoT). Eles obtiveram, em média, pouca melhora com o uso dos 12 atributos do `NetFlow`, um número muito reduzido em comparação ao número de atributos original de cada conjunto de dados, mas obtiveram ganhos significativos com a versão de 43 atributos. Em nosso trabalho também propomos o uso de atributos comuns entre os conjuntos de dados, no entanto, utilizamos um total de 82 atributos de fluxo. Além disso, os autores não avaliam a capacidade de generalização do aprendizado, mas sim o desempenho para cada conjunto de atributos em cada conjunto de dados individualmente.

Na pesquisa de IDSs baseados em AM, comumente busca-se com o processo indutivo partir de resultados particulares – como por exemplo o bom desempenho em um conjunto de dados específico – para chegar a aplicações gerais, como concluir que esse resultado vale para toda uma gama de aplicações. Como exemplo é reportado por [Gupta et al. 2022] que a partir de resultados com os conjuntos de dados UNSW-NB15 e BoT-IoT a abordagem proposta aplica-se para o domínio de sistemas de saúde (*healthcare*), como [Roy et al. 2022] que utiliza-se dos conjuntos de dados NSL-KDD e CICIDS-2017 para concluir a aplicação no domínio de IoT, ou [Ferrag et al. 2021] que utiliza os conjuntos de dados CIC-DDoS2019 e TON-IoT para inferir a aplicação no contexto de agricultura 4.0. No entanto, a generalização do aprendizado em IDS ainda é um desafio.

Em resumo, nosso trabalho destaca-se dos demais por proporcionar uma discussão direcionada à avaliação da capacidade de generalização de modelos de aprendizado de máquina para identificação de ações de *scanning*, levando em consideração fluxos em tráfego TCP em contextos distintos. Outra diferença trata-se da uniformização do conjunto de preditores utilizados, totalizando 82 atributos para treino. Por fim, observa-se a quantificação da importância dos preditores na tomada de decisão. Assim, nosso trabalho avança o estado da arte e proporciona um melhor entendimento do processo de caracterização de ações maliciosas que ocorrem no início de um ataque cibernético.

3. Arcabouço para avaliação

O presente trabalho visa caracterizar os atributos relevantes para a detecção de ataques de *scanning* em sistemas de detecção de intrusão (IDS) baseados em aprendizado de máquina (AM). Esses atributos são descrições *stateful* (fluxo) do tráfego de rede obtidos através de captura do tráfego e posterior processamento através da agregação de suas características a partir de uma 4-tupla representada por endereços IP de origem e destino e respectivas portas resultando em um total de 82 atributos que descrevem determinado tráfego. Na análise *stateful* todo o contexto de comunicação entre dois dispositivos é levado em consideração em contraponto à abordagem *stateless* em que cada troca de informação é considerada isoladamente. Assim, todas as requisições e respostas entre o par cliente e servidor são considerados para compor um atributo de fluxo; como exemplo, o total de pacotes ou bytes transmitidos em uma comunicação definida pela 4-tupla em determinado instante.

Este trabalho considera a importância de se trabalhar com um conjunto comum de atributos para comparação entre diversos conjuntos de dados e demonstra a dificuldade de generalização para contextos diferentes àqueles em que os modelos de AM foram originalmente treinados, deixando claro o desafio para a adoção da abordagem de AM em um cenário real de operação (diverso ao contexto de treinamento). No nosso caso, a decisão por se trabalhar com um ataque em específico na tarefa de classificação deve-se a recomendação de manter o escopo reduzido na aplicação de AM para IDS [Sommer and Paxson 2010].

3.1. Conjuntos de dados (datasets)

As avaliações realizadas neste trabalho cobriram 5 conjuntos de dados distintos, em especial 4 proeminentes na literatura: UNSW-NB15, CICIDS-2017, BoT-IoT e ToN-IoT, e AB-TRAP. Este último é produto do framework AB-TRAP [de Carvalho Bertoli et al. 2021b] no qual são realizados ataques para geração de tráfego malicioso de *scanning*, em uma rede LAN e combinado com tráfego normal recente de forma automatizada. Como o presente trabalho restringe-se à detecção de *scanning*, filtrou-se as outras classes de ataque nos demais conjuntos de dados.

UNSW-NB15 é um composto por tráfego de rede normal e tráfego malicioso sintético gerado pela ferramenta IXIA Perfect Storm. Foi criado no Cyber Range Lab do Australian Centre for Cyber Security (ACCS) em 2015 [Moustafa and Slay 2015]. Para a extração de atributos de fluxo, utilizaram as ferramentas Argus and Bro-IDS resultando em 37 features, além de 12 estatísticas adicionais, totalizando 49 atributos. Ele possuí 10 classes sendo 1 de tráfego benigno e outros 9 ataques, sendo nosso escopo na classe *reconnaissance*.

CICIDS-2017 possuía os ataques mais comuns de sua época [Sharafaldin. et al. 2018], enquanto o tráfego normal foi gerado pelo sistema B-profile, de autoria do próprio grupo Canadian Institute of Cybersecurity (CIC), que emula o comportamento das interações de 25 usuários humanos de serviços de HTTP, HTTPS, FTP, SSH, e protocolos de email. Em adição ao tráfego benigno possuí 8 classes de ataque mas no nosso estudo restringimos para o escopo de detecção de *port scan*. Esse dataset já é disponibilizado com atributos que são

discutidos na seção 3.2.

ToN-IoT foi criado em 2019 e inclui dados de telemetria e serviços de IoT, além de tráfego de rede e log de sistemas operacionais [Alsaedi et al. 2020]. O tráfego foi gerado em uma representação de rede de média escala e heterogênea considerando as camadas de *edge*, *fog* e *cloud*. Possuí 44 atributos que foram extraídos a partir da ferramenta Bro-IDS. Em adição ao tráfego benigno possuí 9 classes de ataque mas no nosso estudo restringimos para o escopo de detecção de *scanning*.

BoT-IoT foi projetado em 2019 um ambiente de tráfego normal e botnets [Koroniotis et al. 2019]. As ferramentas Ostinato e Node-red foram utilizadas para gerar os tráfegos não IoT e IoT enquanto Argus extraiu as 42 features do *dataset* original. Ele possuí 5 classes sendo 1 de tráfego benigno e outros 4 ataques, sendo nosso escopo na classe *reconnaissance*.

AB-TRAP foi o nome dado ao *dataset* criado com o framework homônimo com tráfego emulado para uma rede LAN. Os ataques foram gerados conforme procedimento descrito em Bertoli et al. [de Carvalho Bertoli et al. 2021b] que conforme propõe o arcabouço é agregado ao MAWILab [Fontugne et al. 2010] que provê tráfego atual do backbone da internet em diversos pontos enquanto uma combinação de detectores de anomalia produz rótulos. Neste caso o dataset final é representado em uma classe binária: normal ou scanning.

Cada um dos *datasets* mencionados representa um ambiente diferente de rede, alguns com tipos de serviços distintos. A Tabela 1 apresenta o número de exemplos de cada classe que compõe cada um dos datasets. O balanceamento entre ataques de *port scanning* e tráfego normal para os *dataset* finais foram de 1.74% (AB-TRAP), 1.08% (UNSW-NB15), 6.54% (CICIDS-2017), 1.42% (ToN-IoT) e 97.52% (BoT-IoT). Com exceção do BoT-IoT os demais *datasets* são altamente desbalanceados, sendo a classe de tráfego normal (ou benigno) a de maior representatividade.

Tabela 1. Distribuição dos exemplos por classe em cada dataset

Dataset	Total	Scanning	Normal
AB-TRAP	9.310.859	162.023	9.148.836
CICIDS-2017	2.827.677	158.804	2.271.122
ToN-IoT	5.351.760	36.205	2.515.236
BoT-IoT	13.428.602	3.514.330	89.246
UNSW-NB15	38.987	401	36.675

3.2. Engenharia de atributos (*feature engineering*)

Para garantir o mesmo conjunto de atributos entre os diversos datasets, a ferramenta CICFlowMeterfoi utilizada. O conjunto de atributos produzido pela ferramenta CICFlowMeter a partir dos arquivos `pcap` foi inicialmente descrito em Lashkari et al. [Habibi Lashkari. et al. 2017], contendo 20 métricas temporais de fluxo: o Forward Inter Arrival Time (FIAT) e Backward Inter Arrival Time (BIAT), que são os tempos entre dois pacotes enviados pela origem e destino do fluxo, sendo estas definidas pelo início da transmissão, o Flow Inter Arrival Time (FLOWIAT),

que é o tempo entre a chegada de dois pacotes em qualquer direção, os tempos em que o fluxo permanece nos estados ACTIVE e IDLE, sendo para cada uma dessas medidas gravados seus valores mínimo, máximo, médio e desvio padrão. Além dessas medidas, também são extraídos os valores FLOW_BYTES_S, FLOW_PKTS_S, que são as taxas de bytes e pacotes por segundo, respectivamente, e a duração do fluxo, DURATION. Posteriormente, mais features foram incluídas em Sharafaldin et al.[Sharafaldin. et al. 2018], somando 82 features utilizadas neste trabalho.

O CICIDS-2017 é o único conjunto de dados originalmente disponibilizado com o conjunto de atributos extraídos a partir do CICFlowMeter. O UNSW-NB15 e o AB-TRAP foram re-processados a partir de seus arquivos `pcap` originais. Já para o uso do BoT-IoT e ToN-IoT com o mesmo conjunto de atributos, partiu-se do trabalho de Sarhan et al. [Sarhan et al. 2021] que publicou estes mesmos datasets com as features obtidas pela ferramenta CICFlowMeter. Neste trabalho Sarhan et al. discute a importância de se trabalhar com atributos comuns entre diversos datasets. A escolha do conjunto de atributos e dos *dataset* estudados se deu por facilidade do uso da ferramenta CICFlowMeter e *datasets* já disponíveis nesta condição.

3.3. Pré-processamento

Alguns atributos foram retiradas devido a invariância, identificação de tempo ou de porta e endereço IP, que podem tornar os modelos enviesados para a descrição de seus *testbeds* de origem. Como exemplo, para os conjuntos de dados AB-TRAP e UNBW-NB15, foram retirados os atributos fwd_psh_flags, bwd_psh_flags, fwd_urg_flags, bwd_urg_flags, syn_flag_cnt, rst_flag_cnt, psh_flag_cnt, ack_flag_cnt, urg_flag_cnt, ece_flag_cnt e cwe_flag_count. A porta de destino foi mantida devido a ser uma característica invariante de alguns serviços. Os atributos foram normalizados com o uso do *z-score*, que subtrai a média e mantém uma variância unitária. Para os testes de generalização entre os *datasets*, a normalização feita no *dataset* alvo foi de acordo com o *dataset* utilizado no treinamento do modelo. A obtenção dos resultados foi através de um computador Intel(R) Core(TM) i7-7500U CPU @ 2.70GHz 2.90 GHz com 16GB de RAM, com Python 3.8.8 e os modelos foram treinados e avaliados com o *framework* scikit-learn 0.24.1 e XGBoost 1.5.0.

3.4. Métricas de Aprendizado

Considerando o problema de classificação binária, tráfego normal vs. *scanning*, os quatro resultados possíveis da predição de um modelo podem ser resumidos pela matriz de confusão (Tabela 2). Os resultados podem ser Verdadeiro Positivo (VP), Falso Positivo (FP), Verdadeiro Negativo (VN) ou Falso Negativo (FN), dependendo se, primeiro, foi classificado corretamente ou incorretamente e, segundo, se foi classificado como pertencente à classe ou não. Por exemplo, VN são dados que foram classificados corretamente como não pertencentes à classe de referência. No nosso trabalho a classe *scanning* é a classe positiva.

Tradicionalmente, *acurácia* e *taxa de erro* são as métricas de avaliação mais utilizadas. Elas representam o percentual de acertos no total de classificações e seu reciproco. Essa métrica de avaliação pode apresentar uma falsa sensação de bom desempenho em datasets desbalanceados, como é o caso do nosso problema em questão. Assim, métricas como *F1-score*, *precisão* e *revocação* são mais apropriadas para

Tabela 2. Matriz de Confusão

		Predição do Modelo	
		Positivo	Negativo
Classe real	Positivo	VP	FN
	Negativo	FP	VN

selecionar os modelos treinados, com precisão representando a exatidão e revocação sua completude, enquanto F1-score é a média harmônica ponderada entre eles:

$$\textit{precisão} = \frac{VP}{VP+FP}, \quad \textit{revocação} = \frac{VP}{VP+FN}, \quad \textit{F1-score} = 2\times\frac{\textit{precisão}\times\textit{revocação}}{\textit{precisão}+\textit{revocação}}.$$

Em nosso trabalho, a métrica de referência para as discussões é o F1-score.

4. Análise dos Resultados

O fluxo de trabalho seguido é ilustrado pela Figura 1. Esse fluxo representa a geração do conjunto de dados AB-TRAP e considera que os outros conjuntos de dados estão disponíveis. O passo seguinte consiste na extração de atributos de fluxo comum a todos os conjuntos de dados com o uso da ferramenta CICFlowMeter. Na sequência, são treinados os modelos de aprendizado de máquina, que neste trabalho consideram-se: Decision Tree (DT), Linear Regression (LR), Multilayer Perceptron (MLP), Naïve Bayes (NB), Support Vector Machine (SVM) e Extreme Gradient Boosting (XGB). Estes modelos foram utilizados por apresentarem um bom desempenho nos conjuntos de dados isolados. Técnicas como *deep learning* poderiam ser utilizas, no entanto, para avaliação da generalização de aprendizado, nosso escopo se restringe aos algoritmos de AM clássicos. Por fim, a avaliação do desempenho é feita a partir da métrica F1-score obtida por cada modelo.

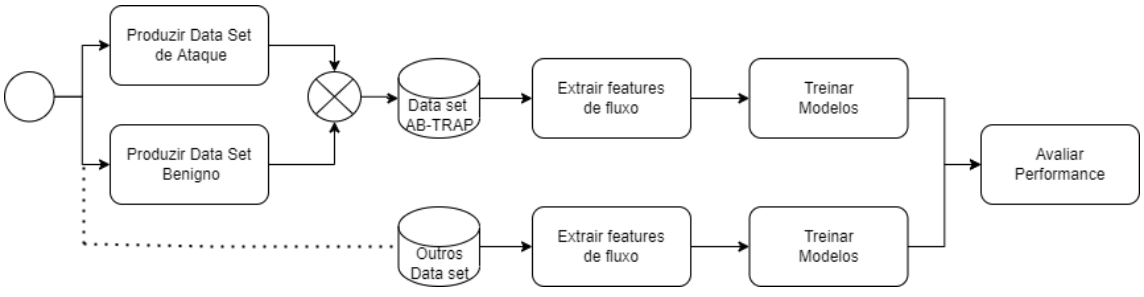

Figura 1. Fluxo de desenvolvimento deste trabalho

4.1. Avaliação em cada conjunto de dados

Para cada combinação de conjunto de dados e algoritmo anteriormente listados, um modelo foi treinado, através de busca uniforme de parâmetros e com F1-score como métrica. Para a busca em modelos DT foram utilizados os critérios de Gini e entropia, profundidade máxima de 5, 10 ou 15, e sem peso entre as classes (balanceado).

Majoritariamente, bons resultados foram obtido por cada combinação de algoritmo e conjunto de dados. Conforme reportado na Tabela 3. Para o modelo DT a busca resultou no uso do critério de entropia, profundidade máxima de 15 e sem peso entre as classes. O modelo MLP foi utilizado com duas camadas ocultas de tamanho 10. Todos os modelos foram treinados considerando a validação cruzada em 10 pastas e estratificadas (*stratified 10-fold*).

Tabela 3. F1-score médio de cada modelo treinado, por dataset

Algoritmo	AB-TRAP	UNBW-NB15	CIC-IDS	ToN-IoT	BoT-IoT
DT	0.980790	0.997500	0.996817	0.911080	0.997001
LR	0.933008	0.732059	0.901173	0.651595	0.992913
MLP	0.970686	0.909874	0.995855	0.704956	0.995024
NB	0.038148	0.714554	0.858466	0.205104	0.980497
SVM	0.929131	0.792698	0.989990	0.612877	0.993067
XGB	0.980094	0.994968	0.996845	0.914894	0.997074

No caso do AB-TRAP obteve-se um F1-score de 0.98 no caso DT e XGB utilizando-se dos 82 atributos de fluxo extraídos com o uso do CICFlowMeter, este resultado é similar ao trabalho original que considerava apenas atributos *stateless* e que reportou um F1-score de 0.97. O mesmo vale para o caso do CIC-IDS, em que o valor de 0.99 é similar ao baseline de 0.98 reportado por [Lucas et al. 2021] e próximo ao obtido pelos autores de 0.999. No entanto, em nossa avaliação nenhuma técnica mais elaborada como *stacking* foi utilizada. A opção por trabalhar com atributos de fluxos do CICFlowMeter se mostrou vantajoso para o caso do BoT-IoT com um desempenho superior ao reportado por [Bochie et al. 2020] que utilizou aprendizado profundo para obter um F1-score máximo de 0.95. O mesmo vale para os autores do conjunto de dados ToN-IoT que reportaram um F1-score de 0.75 no caso multiclasse e 0.88 como classificação binária, com o conjunto de atributos baseados no CICFlowMeter chegamos ao F1-score de 0.91. Mesmo trabalhando com seleção de atributos no conjunto de atributos originais do UNSW-NB15 [Kasongo and Sun 2020], o uso das atributos de fluxo utilizadas no nosso caso apresentou um F1-score superior (0.99 vs. 0.90) sem exercitar a possibilidade de seleção de atributos. Em resumo, a maior quantidade de atributos de fluxo utilizados neste trabalho apresentou desempenho compatível de F1-score com os resultados reportados utilizando-se das atributos originais de cada conjuntos de dados, com exceção do CICIDS-2017 que já possui os mesmos atributos, permitindo a adequada comparação entre os erros de treinamento e de generalização quando aplicado à outro conjunto de dados dissimilar.

4.2. Capacidade de Generalização

Para a avaliação de capacidade de generalização, os modelos treinados em determinado conjunto de dados estimaram as classes em cada um dos outros 4 conjuntos de dados utilizados neste trabalho. Conforme nosso problema de interesse, foi feita classificação binária utilizando somente ataques de *scanning* e tráfego benigno, sem balanceamento dos conjunto de dados, e sem seleção prévia de atributos. Os casos de interesse foram explorados, observando as diferentes métricas disponíveis, para avaliar o desempenho obtido. Para a avaliação cruzada entre os conjuntos de dados,

Tabela 4. Resumo de f1-score médio de cada modelo treinado após undersampling, por combinação de data set origem e alvo. Valores superiores a 0.5 estão em destaque.

Conjunto de dados		Algoritmo					
Treinado em	**Avaliado em**	**DT**	**LR**	**MLP**	**NB**	**SVM**	**XGB**
AB-TRAP	UNSW-NB15	0.019	0	0.006	0.013	0	0.018
	CICIDS-2017	0.001	0.193	0.151	0.165	0.240	0.098
	ToN-IoT	0.037	0	0	0	0	0.033
	BoT-IoT	0.076	0.378	0.786	0.378	0.677	0.214
BoT-IoT	AB-TRAP	0	0	0	0	0	0
	UNSW-NB15	0.001	0.020	0.072	0.278	0.187	0.165
	CICIDS-2017	0.209	0.010	0	0	0.238	0.338
	ToN-IoT	0.095	0.030	0.028	0	0.029	0.088
CICIDS-2017	AB-TRAP	0	0	0	0	0	0
	UNSW-NB15	0	0	0	0	0	0
	ToN-IoT	0.007	0.028	0.028	0	0.028	0.002
	BoT-IoT	0.001	0.001	0	0	0.002	0
UNSW-NB15	AB-TRAP	0	0	0	0	0	0
	CICIDS-2017	0.014	0.043	0	0.001	0.168	0.022
	ToN-IoT	0.037	0.014	0	0	0.014	0.037
	BoT-IoT	0.510	0.846	0.008	0.742	0.941	0.504
ToN-IoT	AB-TRAP	0	0	0	0	0	0
	UNSW-NB15	0.078	0	0.021	0	0	0
	CICIDS-2017	0.006	0.010	0.147	0	0	0.011
	BoT-IoT	0.185	0.629	0.351	0	0.029	0.113

a fim de avaliar sua capacidade de generalização, os resultados obtidos de *F1-score* são apresentados na tabela 4. Nela, estão destacados os valores que obtiveram nota maior que 0.5, a fim de restringir nossa análise nos casos *a priori* mais promissores.

Os resultados reportados na Tabela 4 demonstram a grande dificuldade de generalizar um modelo treinado em um conjunto de dados específico quando testado em outros conjuntos de dados. No nosso caso, ao manter o escopo apenas em ataques de *scanning* buscou-se reduzir a influência dos demais tipos de ataque. Apesar disso, os resultados expõem um grande desafio para a pesquisa de IDS baseados em AM, uma vez que as pesquisas nesta área buscam normalmente generalizar o resultado obtido a partir de um conjunto de dados específico para as mais diversas aplicações conforme os trabalhos relacionados avaliados.

O BoT-IoT foi o único dos *datasets* avaliados em que modelos treinados em outros conjuntos de dados (AB-TRAP, UNSW-NB15 e ToN-IoT) apresentaram um F1-score maior que 0.5. Um ponto de destaque é que o BoT-IoT é o conjunto de dados com o maior desbalanceamento para a nossa classe de interesse (*scanning*) representado por 97.52% do total de exemplos, assim o balanceamento dos conjunto de dados é um ponto importante a ser exercitado nos desdobramentos deste trabalho.

Tabela 5. Features ordenadas por importância para cada modelo treinado na detecção de scanning apresentados na tabela 3. Atributos do CICFlowMeter.

Dataset	DT	LR	MLP	SVM	XGB
AB-TRAP	(67%) bwd_pkts_s	(9%) flow_duration	(8%) dst_port	(8%) subflow_bwd_byts	(46%) bwd_pkts_s
	(7%) bwd_header_len	(6%) subflow_bwd_byts	(7%) pkt_len_std	(8%) totlen_bwd_pkts	(18%) pkt_len_max
	(6%) init_fwd_win_byts	(6%) tot_bwd_pkts	(6%) fin_flag_cnt	(5%) bwd_header_len	(10%) bwd_header_len
	(5%) flow_duration	(6%) totlen_bwd_pkts	(5%) init_fwd_win_byts	(3%) flow_iat_min	(8%) totlen_bwd_pkts
	(3%) dst_port	(6%) subflow_bwd_pkts	(4%) flow_duration	(2%) active_min	(4%) protocol
	(3%) fwd_seg_size_min	(4%) bwd_header_len	(4%) fwd_pkt_len_std	(2%) pkt_len_min	(3%) pkt_len_std
	(2%) pkt_len_var	(4%) down_up_ratio	(4%) pkt_len_min	(2%) bwd_pkt_len_mean	(2%) init_fwd_win_byts
	(2%) totlen_fwd_pkts	(3%) flow_iat_std	(3%) bwd_pkt_len_std	(2%) flow_iat_std	(1%) fwd_pkt_len_min
	(1%) fwd_pkt_len_min	(3%) flow_iat_max	(3%) flow_iat_std	(2%) fwd_pkt_len_max	(1%) fin_flag_cnt
	(1%) fin_flag_cnt	(2%) idle_max	(3%) flow_iat_min	(2%) idle_max	(1%) totlen_fwd_pkts
UNBW-NB15	(88%) fin_flag_cnt	(7%) dst_port	(5%) idle_max	(3%) pkt_len_var	(28%) bwd_pkt_len_mean
	(9%) dst_port	(6%) bwd_pkt_len_max	(5%) active_std	(2%) dst_port	(27%) dst_port
	(3%) pkt_size_avg	(5%) pkt_len_var	(4%) fwd_pkt_len_std	(2%) flow_iat_mean	(22%) fwd_pkt_len_mean
	(0%) idle_max	(5%) fin_flag_cnt	(4%) fwd_iat_max	(2%) flow_iat_max	(10%) flow_duration
	(0%) bwd_iat_tot	(5%) fwd_act_data_pkts	(4%) flow_iat_max	(2%) idle_max	(6%) flow_byts_s
	(0%) bwd_iat_max	(5%) bwd_pkt_len_std	(3%) dst_port	(2%) idle_min	(2%) fin_flag_cnt
	(0%) bwd_iat_min	(4%) idle_min	(3%) idle_std	(2%) fin_flag_cnt	(2%) fwd_pkt_len_max
	(0%) bwd_iat_mean	(4%) flow_iat_mean	(3%) active_max	(2%) active_max	(1%) fwd_iat_max
	(0%) bwd_iat_std	(4%) down_up_ratio	(3%) bwd_iat_std	(2%) fwd_iat_tot	(1%) pkt_len_mean
	(0%) down_up_ratio	(3%) idle_std	(3%) bwd_iat_tot	(2%) bwd_pkt_len_std	(0%) down_up_ratio
CICIDS-2017	(50%) subflow_fwd_byts	(5%) flow_iat_min	(18%) fwd_act_data_pkts	(6%) fwd_act_data_pkts	(50%) totlen_fwd_pkts
	(45%) flow_byts_s	(5%) fwd_pkt_len_max	(14%) tot_fwd_pkts	(5%) subflow_fwd_pkts	(32%) flow_byts_s
	(3%) psh_flag_cnt	(4%) fwd_pkt_len_mean	(14%) subflow_fwd_pkts	(5%) tot_fwd_pkts	(11%) psh_flag_cnt
	(0%) fwd_seg_size_avg	(4%) fwd_pkt_len_std	(11%) tot_bwd_pkts	(4%) pkt_len_mean	(1%) pkt_len_min
	(0%) down_up_ratio	(4%) fwd_seg_size_avg	(11%) subflow_bwd_pkts	(4%) tot_bwd_pkts	(1%) pkt_size_avg
	(0%) ece_flag_cnt	(4%) flow_iat_mean	(2%) fwd_pkts_s	(4%) subflow_bwd_pkts	(0%) down_up_ratio
	(0%) cwe_flag_count	(3%) fwd_pkts_s	(1%) pkt_size_avg	(2%) flow_iat_min	(0%) ece_flag_cnt
	(0%) urg_flag_cnt	(3%) fin_flag_cnt	(1%) fwd_iat_min	(2%) flow_iat_max	(0%) cwe_flag_count
	(0%) ack_flag_cnt	(3%) flow_duration	(1%) subflow_fwd_byts	(2%) pkt_len_max	(0%) urg_flag_cnt
	(0%) dst_port	(3%) tot_fwd_pkts	(1%) bwd_iat_std	(2%) flow_iat_mean	(0%) ack_flag_cnt
ToN-IoT	(44%) fwd_header_len	(24%) dst_port	(5%) bwd_pkt_len_max	(3%) psh_flag_cnt	(19%) bwd_pkt_len_min
	(29%) dst_port	(12%) pkt_len_min	(5%) rst_flag_cnt	(3%) bwd_header_len	(15%) dst_port
	(14%) idle_mean	(4%) bwd_byts_b_avg	(5%) bwd_iat_min	(3%) fwd_header_len	(14%) init_bwd_win_byts
	(4%) bwd_iat_max	(4%) totlen_bwd_pkts	(4%) pkt_len_var	(2%) dst_port	(9%) bwd_pkt_len_std
	(3%) fwd_pkt_len_max	(4%) protocol	(3%) bwd_iat_tot	(2%) flow_iat_mean	(7%) init_fwd_win_byts
	(2%) init_fwd_win_byts	(3%) fwd_iat_tot	(3%) pkt_len_max	(2%) bwd_byts_b_avg	(7%) bwd_iat_max
	(1%) protocol	(3%) pkt_size_avg	(3%) bwd_pkt_len_std	(2%) down_up_ratio	(5%) bwd_iat_std
	(1%) fwd_iat_max	(3%) totlen_fwd_pkts	(3%) syn_flag_cnt	(2%) ack_flag_cnt	(4%) fwd_header_len
	(0%) pkt_len_var	(3%) flow_iat_min	(3%) dst_port	(2%) rst_flag_cnt	(4%) bwd_pkts_s
	(0%) fin_flag_cnt	(3%) pkt_len_var	(3%) psh_flag_cnt	(2%) syn_flag_cnt	(3%) idle_mean
BoT-IoT	(39%) syn_flag_cnt	(8%) init_bwd_win_byts	(24%) bwd_header_len	(9%) tot_bwd_pkts	(69%) protocol
	(17%) dst_port	(8%) bwd_iat_tot	(7%) tot_bwd_pkts	(9%) subflow_bwd_pkts	(10%) fwd_pkt_len_min
	(11%) flow_iat_min	(5%) tot_fwd_pkts	(7%) subflow_bwd_pkts	(8%) bwd_header_len	(3%) init_bwd_win_byts
	(10%) protocol	(5%) subflow_fwd_pkts	(4%) fwd_header_len	(8%) fwd_act_data_pkts	(2%) syn_flag_cnt
	(7%) bwd_header_len	(5%) fwd_header_len	(3%) tot_fwd_pkts	(5%) fwd_header_len	(2%) pkt_len_max
	(4%) init_bwd_win_byts	(4%) fwd_act_data_pkts	(3%) subflow_fwd_pkts	(3%) tot_fwd_pkts	(1%) dst_port
	(3%) fwd_iat_min	(4%) flow_duration	(3%) down_up_ratio	(3%) subflow_fwd_pkts	(1%) flow_iat_min
	(3%) subflow_fwd_pkts	(4%) fwd_iat_tot	(3%) fwd_pkt_len_max	(3%) fwd_iat_std	(1%) fwd_act_data_pkts
	(2%) fwd_pkts_s	(4%) bwd_iat_std	(3%) init_bwd_win_byts	(2%) flow_iat_mean	(1%) ack_flag_cnt
	(1%) flow_iat_mean	(3%) bwd_iat_max	(2%) flow_iat_std	(2%) subflow_bwd_byts	(1%) rst_flag_cnt

4.3. Importância dos atributos

A análise de importância dos atributos é uma forma de interpretar o aprendizado obtido. Permite a discussão com especialistas ou mesmo entender características do tráfego daquela rede. Essas características representadas pelos atributos de fluxo podem mudar ao longo do tempo (*concept drift*), enquanto as características dos ataques e os atributos que os descrevem tendem a se manter com o mesmo comportamento quando técnicas de evasão não são utilizadas. A Tabela 5 apresenta os 10 atributos de maior importância para cada um dos modelos treinados para a tarefa de classificação binária entre tráfego normal e de *scanning* em cada um dos conjuntos de dados criados com o conjunto de 82 atributos obtidos com o CICFlowMeter. Uma análise de cada algoritmo permite concluir que a LR, MLP e SVM normalmente não dão um peso muito grande para poucos atributos, ao contrário do que vemos para

os modelos baseados em árvores como a DT e também o XGB que utiliza-se da técnica de *boosting*. O cálculo de importância dos atributos para DT e XGB foram extraídos diretamente do processo de treinamento sendo atributos disponíveis como parte dos modelos finais treinados. Para o caso da LR, MLP e SVM a obtenção da importância dos atributos foi obtida a partir dos parâmetros do modelo final. No modelo LR, o valor absoluto dos coeficientes do preditor foram utilizados como importância. Para a MLP, uma aproximação com o somatório dos valores absolutos de todos os caminhos na rede foi utilizado, com cada caminho aproximado pelo produtório dos pesos da rede. Finalmente, para o modelo SVM, a importância se baseou na distância mínima da curva de decisão do vetor que possui valor unitário no atributo avaliado e zero em todos os outros. Isto se assemelha a calcular o vetor de importância através da função de decisão do modelo com a matriz identidade como argumento. Após calculadas as importâncias de todos os atributos, os mesmos foram normalizados pelo somatório total das importâncias para o respectivo modelo.

Uma análise de cada contexto (conjunto de dados) com o objetivo de avaliar o aprendizado de cada algoritmo permite concluir que existem alguns tipos de atributos que são explicativos dos ataques de *scanning* analisados nos conjuntos de dados. Um grupo de atributos são aqueles associados as flags do protocolo TCP (*fin_flag_cnt*, *syn_flag_cnt*, *psh_flag_cnt*) que podem representar um excesso de pacotes que são utilizados em ataques como o SYN, FIN ou XMAS scan e trabalham com a manipulação dessas flags em pacotes construídos para explorar a resposta dos sistemas a esses pacotes. Um outro atributo que apresenta grande ocorrência nos diversos contextos é o *dst_port* que teve a decisão de ser mantido em nossa análise e que no caso dos conjuntos de dados analisados pode acabar por ser um descritivo dos testbeds utilizados, o mesmo entendimento vale para o atributo *protocol*. Merecem destaque também os atributos associados ao comprimento dos pacotes e a quantidade de bytes trafegados. Para esses dois últimos conjuntos de atributos, uma análise do tráfego normal utilizado em cada testbed se faz necessário. A incapacidade de aprender atributos com grande importância e comum aos diversos *datasets* reforçam a dificuldade de generalização de aprendizado para a classificação de ataques de *scanning* conforme visto na Tabela 4 e os resultados obtidos na Tabela 3 podem ser entendidos com bom desempenho em descrever os conjuntos de dados.

5. Conclusão

Este trabalho apresentou um estudo envolvendo cinco modelos (*decision tree*, *logistic regression*, *multilayer perceptron*, *support vector machine* e *eXtreme Gradient Boost* aplicados ao problema de classificação de pacotes maliciosos de scan em fluxos TCP. Ao uniformizar os atributos preditores por meio do CICFlowMeter em cinco conjuntos de dados diferentes pode-se obter valores superiores ao estado da arte. Outro ponto para discussão diz respeito à dificuldade de generalização do aprendizado em sistemas de detecção de intrusão (IDS). Assim, destaca-se que modelos treinados em um contexto (conjunto de dados) notoriamente não apresentam um bom desempenho em outro. E esse é um resultado importante pois deixa claro o desafio na definição de preditores contidos em fluxos de tráfego de rede. Por fim, uma contribuição adicional para a pesquisa de IDS é a disponibilização do conjunto de dados UNSW-NB15 (amplamente utilizado) com o conjunto de atributos gerados a partir do CICFlowMeter,

somando-se aos já disponíveis CIC-IDS2017, BoT-IoT e ToN-IoT. Como trabalhos futuros, destaca-se a generalização do aprendizado entre os diversos conjuntos de dados compostos do mesmo conjunto de atributos. Para isso, consideramos a adoção de técnicas de seleção de atributos (otimização considerando os demais conjuntos de dados), utilização dos demais conjuntos de dados como parte do processo de validação, balanceamento dos conjuntos de dados, comitê de modelos, aprendizado federado e aprendizado não-supervisionado. O código-fonte para reprodução do trabalho está disponível em: `https://github.com/c2dc/aBFF-sbseg2022`.

Referências

Al-Riyami, S., Coenen, F., and Lisitsa, A. (2018). A re-evaluation of intrusion detection accuracy: alternative evaluation strategy. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2195–2197.

Al-Sarawi, S., Anbar, M., Abdullah, R., and Al Hawari, A. B. (2020). Internet of things market analysis forecasts, 2020–2030. In *2020 Fourth World Conference on Smart Trends in Systems, Security and Sustainability (WorldS4)*, pages 449–453.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., and Anwar, A. (2020). Ton_iot telemetry dataset: A new generation dataset of iot and iiot for data-driven intrusion detection systems. *IEEE Access*, 8:165130–165150.

Apruzzese, G., Pajola, L., and Conti, M. (2022). The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, pages 1–1.

Bochie, K., Gonzalez, E. R., Giserman, L. F., Campista, M. E. M., and Costa, L. H. M. (2020). Detecção de ataques a redes iot usando técnicas de aprendizado de máquina e aprendizado profundo. In *Anais do XX Simpósio Brasileiro em Segurança da Informação e de Sistemas Computacionais*. SBC.

Buczak, A. L. and Guven, E. (2016). A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Communications Surveys Tutorials*, 18(2):1153–1176.

Catillo, M., Pecchia, A., Rak, M., and Villano, U. (2021). Demystifying the role of public intrusion datasets: a replication study of dos network traffic data. *Computers & Security*, page 102341.

de Carvalho Bertoli, G., Júnior, L. A. P., Verri, F. A. N., dos Santos, A. L., and Saotome, O. (2021a). Bridging the gap to real-world for network intrusion detection systems with data-centric approach. *NeurIPS - Data-centric AI Workshop*.

de Carvalho Bertoli, G., Pereira Junior, L. A., Saotome, O., Dos Santos, A. L., Verri, F. A. N., Marcondes, C. A. C., Barbieri, S., Rodrigues, M. S., and Parente De Oliveira, J. M. (2021b). An end-to-end framework for machine learning-based network intrusion detection system. *IEEE Access*, 9:106790–106805.

Ferrag, M. A., Shu, L., Djallel, H., and Choo, K.-K. R. (2021). Deep learning-based intrusion detection for distributed denial of service attack in agriculture 4.0. *Electronics*, 10(11).

Fontugne, R., Borgnat, P., Abry, P., and Fukuda, K. (2010). MAWILab: Combining Diverse Anomaly Detectors for Automated Anomaly Labeling and Performance Benchmarking. In *ACM CoNEXT '10*, Philadelphia, PA.

Gupta, L., Salman, T., Ghubaish, A., Unal, D., Al-Ali, A. K., and Jain, R. (2022). Cybersecurity of multi-cloud healthcare systems: A hierarchical deep learning approach. *Applied Soft Computing*, page 108439.

Habibi Lashkari., A., Draper Gil., G., Mamun., M. S. I., and Ghorbani., A. A. (2017). Characterization of tor traffic using time based features. In *Proceedings of the 3rd International Conference on Information Systems Security and Privacy - ICISSP,*, pages 253–262. INSTICC, SciTePress.

Kasongo, S. M. and Sun, Y. (2020). Performance analysis of intrusion detection systems using a feature selection method on the unsw-nb15 dataset. *Journal of Big Data*, 7(1):1–20.

Kenyon, A., Deka, L., and Elizondo, D. (2020). Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. *Computers & Security*, 99:102022.

Koroniotis, N., Moustafa, N., Sitnikova, E., and Turnbull, B. (2019). Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-iot dataset. *Future Generation Computer Systems*, 100:779–796.

Layeghy, S. and Portmann, M. (2022). On generalisability of machine learning-based network intrusion detection systems.

Lucas, T. J., da Costa, K. A., Moraes, E. A., Júnior, P. R. H., and das Neves, M. J. (2021). Stacking-based committees para detecção de ataques em redes de computadores-uma abordagem por exaustão. In *Anais do XXXIX Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 644–657.

Moustafa, N. and Slay, J. (2015). Unsw-nb15: a comprehensive data set for network intrusion detection systems (unsw-nb15 network data set). In *2015 military communications and information systems conference (MilCIS)*, pages 1–6. IEEE.

Roy, S., Li, J., Choi, B.-J., and Bai, Y. (2022). A lightweight supervised intrusion detection mechanism for iot networks. *Future Generation Computer Systems*, 127.

Sarhan, M., Layeghy, S., and Portmann, M. (2021). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications*, pages 1–14.

Sharafaldin., I., Habibi Lashkari., A., and Ghorbani., A. A. (2018). Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy - ICISSP,*, pages 108–116. INSTICC, SciTePress.

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*, pages 305–316.