Classificação do Tráfego de Rede por Meio do Fluxo de Rede Utilizando Algoritmos de Aprendizado de Máquina para Detecção de Anomalia

Welton T. M. Sousa[1], **Carlos A. Silva**[1]

[1]Departamento de Informática – Instituto Federal de Minas Gerais (IFMG)
CEP 34.590-390 – Sabará, MG – Brasil

weltonthiago@gmail.com, carlos.silva@ifmg.edu.br

Abstract. *The classification of network traffic through the network flow holds great in the context of cyber attacks. Several segments of society are for these efforts, which represent significant financial. Efficient methods such as preventive and firewall not always identify anomalies in the network. Reason for these circumstances, this article proposes the study and application of seven machine learning algorithms with the use of feature selection of network application traffic, prioritizing automation in anomalous flow identification, contributing to obtain metrics helping decision making by network administrators.*

Resumo. *A classificação do tráfego de rede por meio do fluxo de rede possui grande relevância no contexto dos ataques cibernéticos. Diversos segmentos da sociedade são afetados por esses ataques, os quais podem representar perdas financeiras significativas. Além disso, os métodos convencionais como antivírus e firewall nem sempre conseguem identificar anomalias na rede. Motivado por estas circunstâncias, este artigo propõe o estudo e aplicação de sete algoritmos de aprendizado de máquina com a utilização de seleção de atributos na classificação do tráfego de rede por meio do fluxo de rede, primando a automatização na identificação de fluxo anômalo, contribuindo para a obtenção de métrica no auxílio à tomada de decisão por administradores de redes.*

1. Introdução

A ascensão à informação não autorizada configura um grave problema nas organizações, isso ocorre em alguns casos devido a ações de *crackers*, *malwares* ou *ransomware*, sendo que este último, um sequestrador de dados, solicita um resgate em dinheiro para restaurar os mesmos, cujo problema ocasiona indisponibilidade no acesso a informação. Em um contexto competitivo em que a informação é o principal fator na tomada de decisão, seja em instituições públicas ou privadas, é importante assegurar disponibilidade, autenticidade e integridade nos acessos aos recursos computacionais, *on-premise* ou *cloud*, por meio das redes de computadores.

Segundo o relatório anual da *IBM security*[1], publicado em 24 de fevereiro de 2021, os ataques cibernéticos às redes corporativas ampliaram consideravelmente nos segmentos de saúde, manufaturas e energia em relação ao ano de 2020, decorrente da exploração de vulnerabilidades. Os principais ciberataques foram *phishing*, *ransomware* e *DDoS*

[1]https://www.ibm.com/blogs/ibm-comunica/ibm-security-ataques-ciberneticos/

com a finalidade de parar os serviços por um determinado tempo, conforme abordado em [Pranggono and Arabo 2021], por consequência, ocasionando indisponibilidade e perdas financeiras [Chigada and Madzinga 2021].

Os administradores de redes são responsáveis por atuar no gerenciamento e projeto dos recursos computacionais das organizações, garantindo o acesso ao conteúdo e minimizando risco à segurança da informação. Nesse sentido, o monitoramento de rede é indispensável para a obtenção de métricas de desempenho, ou seja, analisar o tráfego de rede é importante na identificação de comportamentos anômalos.

Neste trabalho buscou-se avaliar a efetividade na detecção de anomalia no tráfego de rede através do fluxo fundamentado na base de dados *offline* **UNSW-NB15**, realizando estudo e análise da eficácia da classificação dos algoritmos de aprendizado de máquina por intermédio da avaliação das métricas de desempenho. Espera-se, portanto, que esse instrumento de reconhecimento do comportamento incomum no tráfego de rede *offline* possa auxiliar no desenvolvimento de uma futura ferramenta na tomada de decisão pelos administradores de redes.

Desse modo, para alcançar o objetivo central desse artigo, o texto encontra-se organizado da seguinte forma: Na seção 1 é feita a introdução da temática abordada no trabalho. Na seção 2, relevantes trabalhos da literatura relacionados ao tema de pesquisa são apresentados. Na seção 3, a metodologia empregada é descrita. Na seção 4 é detalhada cada etapa do desenvolvimento realizado, desde a base de dados e o seu tratamento até o implementação e aplicação dos algoritmos propostos, bem como as análises e discussões a respeito dos resultados obtidos. E por fim, na seção 5 são apresentadas as conclusões finais.

2. Trabalhos Relacionados

Em seu estudo [Moustafa and Slay 2015a] os autores utilizaram a técnica de seleção de *features*, *Association Rule Mining*, cujo método consiste na avaliação de dois ou mais *features* da base de dados, agrupando as melhores na etapa de pré-processamento, diminuindo o número de *features*. Após a seleção das melhores *features* conduziu-se o treinamento do modelo com o algoritmo *Naive Bayes* cujas métricas de desempenho obtidas foram acurácia 37,5% e FAR 62,6%. Utilizando o algoritmo *Expectation–Maximization* foi obtida a acurácia 23,8% e FAR 75,8% para classificação binária da base de dados UNSW-NB15 *Testing*, ou seja, indicando fluxo normal e anômalo sem levar em consideração os tipos de ciberataque.

Segundo [Moustafa and Slay 2016] em seu trabalho, para a seleção dos atributos na base de dados UNSW-NB15 foram utilizadas técnicas de análise estatística e correlação. Posteriormente realizou-se o treinamento com os algoritmos de aprendizado de máquina *Decision Tree* com respectiva acurácia de 85,5% e FAR 15,8%, *Linear Regression* com acurácia 83,1% e FAR 18,5%, *Naive Bayes* com acurácia 82,1% e FAR 18,5%, *Artificial Neural Network* com acurácia 81,3% e FAR 21,1% e *Expectation–Maximization* com acurácia 78,5% e FAR 23,8%.

Em [Janarthanan and Zargari 2017] é proposto a utilização do WEKA, no qual vários métodos e algoritmos são implementados, utilizando especificamente os métodos: *CfsSubsetEval*, *GreedyStepwise*, *InfoGainAttributeEval* e *Ranker* em conjunto com o al-

goritmo *Random Forest* para seleção de atributos (*features*) na base de dados UNSW-NB15. As métricas para avaliação da seleção de atributos foram acurácia "Instância Classificada Corretamente" com 75,7% e κ (coeficiente *kappa* de Cohen) com 82,9% utilizada para mensurar a concordância entre a categorização predita e a esperada na base de dados *Training* e acurácia "Instância Classificada Corretamente" com 76,4% e κ com 81,6% na base de dados *Testing*.

[Jing and Chen 2019] discorre sobre a classificação binária de conjuntos de dados UNSW-NB15, ou seja, a classificação do tráfego baseada em fluxos de rede normais e anômalos implementada na linguagem Java, realizando a transformação dos atributos fundamentada na escala logarítmica, na etapa de pré-processamento. Realizou-se a implementação do algoritmo *Support Vector Machine* com o parâmetro de *kernel RBF* e utilizou-se a validação cruzada para treinamento do modelo, alcançando os seguintes resultados na base de dados *Testing*, acurácia 85,9% e FAR 15,3% .

[Sarhan et al. 2020] utilizou a abordagem no pré-processamento de eliminação dos atributos identificadores do fluxo de rede como IP de origem, IP de destino, *sttl*, *dttl* e *ct_state_ttl* na base de dados UNSW-NB15. Aplicou-se a técnica de transformação *Min-Max Scaling* para dimensionar os atributos da base de dados. Por fim, conduziu-se na utilização do algoritmo *Ensemble Extra Trees Classifier* constituído de 50 estimadores para criação das árvores de decisão. A classificação do fluxo binário, alicerçado na base de dados *Testing* sucedeu nas seguintes métricas de desempenho acurácia 99,2%, AUC 95,4%, *f1-score* 92,0%, DR 91,2% e FAR 0,3%.

Embora a utilização dos algoritmos de aprendizado de máquina possam apresentar falsos positivos, o presente artigo propõe a utilização de técnicas de pré-processamento com a seleção de atributos e transformação nos dados, realizando assim, o treinamento do modelo, levando em consideração as métricas de classificação, objetivando a minimizar FAR e maximizar as métricas acurácia, precisão, sensibilidade, *f1-score* e AUC.

3. Metodologia

O presente trabalho caracteriza-se como pesquisa aplicada de caráter descritivo, que visa estudar e analisar a eficácia de algoritmos de aprendizado de máquina supervisionados com a implementação dos seguintes métodos: *Recursive Feature Elimination* - RFE com *LinearSVC*, *f_classif*, *chi2* e *Random Forest* - RF, para a realização da seleção de *features*, ou seja, minimizar o número de atributos no treinamento do modelo supervisionado. Foram implementados os seguintes algoritmos supervisionados: *K Nearest Neighbor* - KNN, *Logistic Regression* - LR, *Support Vector Machine* - SVM, *Naive Bayes* - NB, *Decision Tree* - DT, *Random Forest* - RF e *Gradient Boosting* - GB para a classificação do tráfego de rede com identificação de anomalia através do fluxo de rede.

Nesse sentido conduziu-se utilizando o método hipotético dedutivo, com levantamento dos dados secundários e revisão bibliográfica. A base de dados utilizada foi desenvolvida pela Universidade de Nova Gales do Sul em Sydney [Moustafa and Slay 2015b], considerando o tráfego de rede normal e anômalo. Portanto, a apresentação dos resultados é quantitativa mediante análise dos resultados pelas métricas: acurácia, precisão, sensibilidade, *f1-score*, AUC e FAR, observado o contexto e objetivos deste trabalho.

4. Desenvolvimento

Para o desenvolvimento do trabalho foram utilizadas as seguintes ferramentas: *Google Colaboratory* (ambiente de programação), *Google Drive* (armazenamento da instância), linguagem *Python* v3.7, além das bibliotecas *pandas*, *scikit-learn*, *numpy* e *matplotlib*.

A seguir são detalhadas as etapas do desenvolvimento. Após importar as bases de dados (*Training* e *Testing*) no *Google Colab*, foram realizadas as seguintes etapas: análise exploratória de dados, exclusão de *features*, transformação dos dados com a utilização do *One Hot Enconding*, seleção de *features*, normalização dos dados, realização do treinamento e avaliação dos modelos mediante os algoritmos implementados, finalizando a predição do modelo treinado na base de dados *Testing* e obtenção das métricas de desempenho da classificação.

O modelo de classificação *offline* proposto neste trabalho, pode ser resumido pela Figura 1.

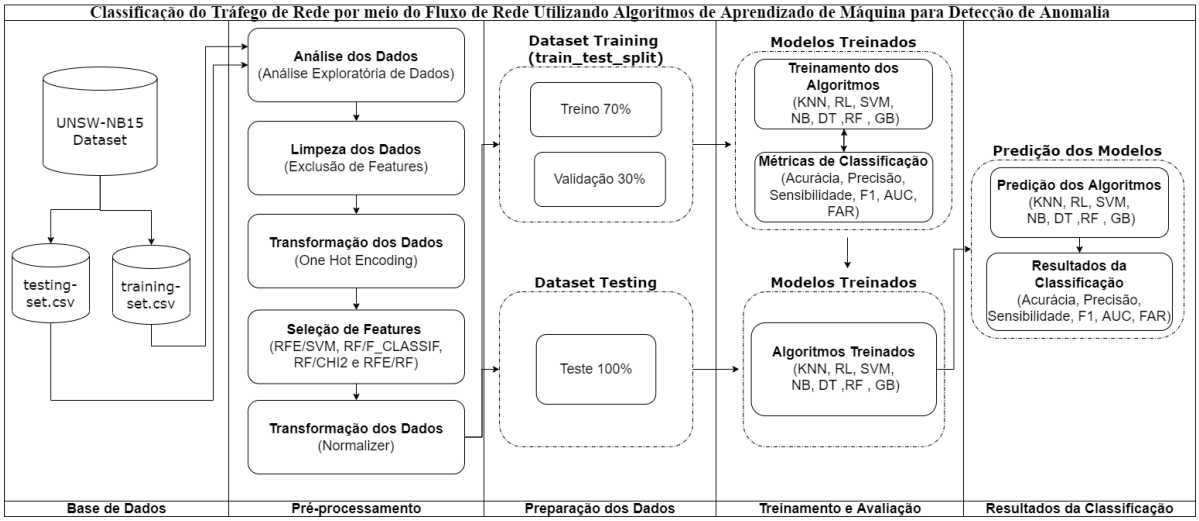

Figura 1. Modelo de Classificação *offline*.

4.1. Base de dados

A base de dados (*dataset*) UNSW-NB15[2] foi desenvolvida com o objetivo de reproduzir o cenário atual das redes, tendo em vista que os *datasets* disponíveis na literatura foram gerados a cerca de uma década, na qual o comportamento da rede, seja tráfego normal ou anômalo é diferente. Para simular o tráfego de rede normal e anômalo, o Centro Australiano de Segurança Cibernética desenvolveu a ferramenta *IXIA PerfectStorm no Cyber Range Lab* para criar de forma sintética, atividades de comportamento normal e de nove formas de ciberataque obtidas através do site *Common Vulnerabilities and Exposures*[3], no qual funciona como uma base de dados referente às vulnerabilidades encontradas e exposições relacionadas à segurança da informação. O procedimento de captura bruta de pacotes e armazenamento foi realizada pela ferramenta de análise e captura de tráfego de rede *Tcpdump*[4] exportando a captura no arquivo *PCAP*.

[2]https://research.unsw.edu.au/projects/unsw-nb15-dataset

[3]https://cve.mitre.org/cve/search_cve_list.html

[4]https://www.tcpdump.org/manpages/tcpdump.1.html

Após a geração do arquivo *PCAP*, foi realizado o procedimento de categorização pelas ferramentas *Bro-IDS*[5], no qual é um *Network Intrusion Detection System*, responsável pela análise do tráfego de rede com identificação de ciberataques, e por último o *Argus*[6] incumbido de gerar os fluxos de rede linha a linha com as respectivas categorias anteriores no formato CSV para utilização no *Python*.

A UNSW-NB15 disponibilizou duas bases de dados (*datasets*), o **UNSW_NB15_training-set.csv** para treinamento e avaliação e o **UNSW_NB15_testing-set.csv** para teste, ambos com 45 *features* contendo fluxo de rede catalogados como normal e anômalo.

4.2. Análise e Tratamento dos Dados

A análise exploratória dos dados consiste na identificação do conteúdo dos dados, auxiliando no reconhecimento da dispersão, desvio padrão, variáveis categóricas, correlação entre *features*, identificação de valores faltantes (*missing*), dentre outras técnicas, por consequência melhorar a tomada de decisão na modelagem do problema. Mediante a análise exploratória, dirigiu-se na idealização da estratégia para o tratamento dos dados, ou seja, o pré-processamento. O tratamento dos dados (pré-processamento) implica na manipulação, estruturação e organização, que precede a realização das predições, sendo importante, pois impacta diretamente na qualidade final da análise. No presente trabalho foram destacadas e realizadas três ações específicas nas bases de dados (*Training* e *Testing*):

- Limpeza dos dados com exclusão da *feature* '**id**'.
- Limpeza dos dados com exclusão da *feature* '**attack_cat**', pois ao manter o tipo de ataque nos *datasets* (*Training* e *Testing*) ocorrerá o sobreajuste (*overfitting*) por consequência do vazamento (*data leakage*), em virtude de que os ataques no mundo real não estarão catalogados.
- Utilização do **One Hot Enconding** - OHE nas *features*: '**proto**', '**service**', '**state**', por ser uma *feature* categórica, no qual é necessário converter os dados sem afetar a segmentação equivalente. É criado um *array* com o valor 1 para a *feature* booleana referente a categoria e 0 na *feature* que não existe na categoria, isso em cada linha do fluxo.

Foram realizadas a instalação do pacote **category_encoders** e utilização do método OHE, possibilitando categorização dos atributos mediante a criação do sufixo *underscore*, após o nome de cada *feature*: '**proto_**', '**service_**', '**state_**', para identificar cada valor e a sua respectiva *feature* de origem. Após o procedimento do OHE, o número de *features* nas bases de dados (*Training* e *Testing*) diferem entre si. Em seguida dirigiu-se a verificação dos nomes das *features* e a criação das inexistentes nas respectivas bases de dados, além da inserção do valor **int** "0" para povoar os conjuntos de dados.

4.3. Seleção de Features

Nesta etapa foram realizadas quatro simulações por intermédio dos algoritmos implementados de seleção de *features*, utilizando a base de dados *Training* com o propósito de reduzir o número de 197 *features* após categorização pelo OHE, nesse sentido, refinar

[5]https://bricata.com/blog/what-is-bro-ids/
[6]https://openargus.org/

o treinamento e predição do modelo. O número de 10 *features* foi definido conforme avaliação de três características principais: menor número de *features* visto que impacta diretamente no tempo computacional, maior valor das métricas acurácia e κ (coeficiente *kappa* de Cohen) que estão diretamente relacionadas com o quão acurado o valor obtido pelo modelo está do real.

Nas simulações foram utilizados os algoritmos RFE/RF, com método *SelectKbest*, além dos métodos estatísticos *f_classif* e *Chi2* para a seleção das melhores *features*. As métricas de desempenho utilizadas, foram: menor número de *features* maximizando o valor da acurácia e κ. Em todas as simulações ocorreram duas etapas: a primeira etapa incide sobre o procedimento de separação do treinamento e avaliação utilizando os seguintes parâmetros: *stratify*: esse parâmetro aloca de forma proporcional as classes 0 e 1 no treino e avaliação; *test_size*: 70% treino e 30% avaliação; e *random_state*: com valor numérico 78, para controlar a aleatoriedade. A segunda etapa das simulações consiste na seleção das *features* utilizando métodos computacionais como *LinearSVC*, *SelectKbest* e *Random Forest Classifier*. Para todas as simulações foram obtidas as 10 melhores *features*, sendo que para a segunda simulação, implementando o algoritmo RFE e as duas últimas simulações implementando o algoritmo RF adotou-se o parâmetro para criação de 100 árvores na floresta. Utilizou-se todos os processadores disponíveis em paralelo e considerou o valor numérico 78, para controlar a aleatoriedade das amostras na construção das árvores e a manutenção de sua reprodutibilidade.

A Tabela 1 descreve sucintamente os resultados das simulações e *features* obtidas.

Tabela 1. Resultados das simulações.

Simulação	**Seleção das melhores** *features*	**Teste estatístico**	*features*	**Acurácia**	**κ**
RFE	LinearSVC		'ct_state_ttl', 'label', 'proto_udp', 'proto_arp', 'service_pop3', 'service_ssl',	0,82	0,64
RFC	SelectKbest	f_classif	'service_ssh', 'state_INT', 'rate', 'sttl', 'swin', 'dwin', 'ct_state_ttl'	0,85	0,69
RFC	SelectKbest	Chi2	'sbytes', 'dbytes', 'rate', 'sload', 'dload', 'sinpkt', 'sjit', 'stcpb', 'dtcpb', 'response_body_len'	0,93	0,87
RFE	RFC		'dpkts', 'sbytes', 'dbytes', 'rate', 'dttl', 'ackdat', 'ct_srv_src', 'ct_src_dport_ltm', 'ct_dst_sport_ltm', 'ct_ftp_cmd'	0,97	0,94

Nota-se que a acurácia em todas as simulações apresentam valor superior a 80%, com destaque para a quarta simulação com 97% de acurácia. Os algoritmos utilizados *RFE/RF* obtiveram os melhores desempenhos na seleção das 10 principais *features*, consistindo no modelo de seleção escolhido entre as simulações propostas para utilização no treinamento e análise de desempenho.

4.4. Modelos, Treinamento e Análise de Desempenho

Nesta seção são apresentados os modelos de predição com os algoritmos de aprendizado de máquina propostos para treinamento, além dos resultados mediante cinco medidas de desempenho na base de dados *Testing*.

Devido a similaridade na etapa de predição pelos algoritmos de aprendizado de máquina, conduziu-se exibindo somente o primeiro modelo. Os demais seguem o idêntico procedimento, alterando basicamente o algoritmo e seus respectivos parâmetros e atributos. Para a construção dos modelos de predição são listados cinco passos a saber:

1. Divisão das variáveis preditoras (seleção das *features* obtidas pelo algoritmo *RFE/RF* na etapa anterior) e variável alvo (*target*).
2. Procedimento de separação do treino e avaliação pelo *train_test_split*.
3. Normalização das variáveis preditoras.
4. Instanciação do modelo e treinamento com algoritmo.
5. Predição do modelo treinado com o *dataset Testing* e exibição das métricas de desempenho.

O **passo 1** procede com a divisão das variáveis preditoras e o alvo (*target*), atribuídas respectivamente nas variáveis X e Y. No **passo 2** ocorre a separação do treino e a avaliação pela função *train_test_split*, utilizando respectivamente 70% e 30%. Esta função, presente no *scikit-learn*, divide os dados em conjuntos de treinamento e avaliação. Em seguida conduziu-se realizando a normalização das variáveis preditoras, configurando assim o **passo 3**. Dirigiu-se na utilização em todos os modelos o método de normalização: *Normalizer*. No **passo 4** dirigiu-se a instanciação do modelo e treinamento com algoritmo. As relações entre os modelos podem ser visualizadas na Tabela 2, sendo que para determinados algoritmos foram utilizados parâmetros específicos, como: **KNN** ($n\_neighbors = 5$), **SVC** ($probability = True$), **RF** ($n\_estimators = 1000$), **GB** ($n\_estimators = 1000$). A predição dos modelos treinados com a base de dados *Testing* e a exibição das métricas de desempenho objetivam o **passo 5**.

A Tabela 2 apresenta o desempenho de cada modelo associado ao seu respectivo algoritmo de aprendizado de máquina, considerando as métricas: acurácia, precisão, sensibilidade, *f1-score*, AUC e FAR.

Tabela 2. Desempenho dos modelos treinados.

Modelo	Algoritmo	Medidas de desempenho (%)					
		Acurácia	Precisão	Sensibilidade	F1-score	AUC	FAR
1	K-Nearest Neighbors	90,0	97,0	88,0	92,0	96,0	9,0
2	Logistic Regression	71,0	83,0	71,0	77,0	80,3	30,0
3	Support Vector Machine	80,0	93,0	76,0	84,0	91,0	18,0
4	Naive Bayes	62,0	70,0	78,0	74,0	72,0	47,0
5	Decision Tree	90,0	97,0	88,0	92,0	91,0	9,0
6	Random Forest	91,0	97,0	89,0	93,0	98,0	8,0
7	Gradient Boosting	91,0	98,0	89,0	93,0	98,0	8,0

Os modelos utilizando os algoritmos: KNN, DT, RF e GB obtiveram resultados satisfatórios mediante as métricas acurácia, precisão e AUC, com valores iguais ou superiores a 90%, além de apresentar um valor de FAR abaixo de 10%. Pode-se inferir que o modelo utilizando o algoritmo *Gradient Boosting* pode ser utilizado como alternativa factível para classificação binária do fluxo de rede para identificação de anomalia devido sua precisão de 98% e taxa de alarmes falsos FAR 8%.

5. Conclusão

Conforme apresentado ao longo do artigo, a partir da pesquisa e análise de classificação do tráfego de rede por meio do fluxo utilizando os algoritmos de aprendizado de máquina

para classificação binária, ou seja, fluxo de rede normal e anômalo na base de dados UNSW-NB15, pode-se, então, ratificar sua relevância no reconhecimento de fluxo de rede destoante. A classificação do fluxo de rede é crucial como métrica de desempenho para monitoramento da rede, servindo de subsídio na tomada de decisão pelos administradores de redes. Portanto, dentre os algoritmos de aprendizado de máquina propostos: KNN, RL, NB, SVM, DT, RF e GB para classificação do fluxo, percebe-se que o algoritmo *Gradient Boosting*, obteve as melhores métricas, direcionando-se como alternativa factível para identificação de anomalia.

Comparando os resultados da revisão da literatura com os obtidos no desenvolvimento deste trabalho, nota-se um melhor desempenho em relação à medida de acurácia do método implementado *Random Forest* (91,0%) em relação ao mesmo método apresentado em [Moustafa and Slay 2016] e [Janarthanan and Zargari 2017]. Para a medida de desempenho FAR, os métodos implementados *Naive Bayes*, *Random Forest* e *Logistic Regression* obtiveram melhor desempenho do que os métodos da revisão da literatura. Portanto, mediante os resultados obtidos, os algoritmos de aprendizado de máquina emergem como alternativas no estudo e implantação de futuras ferramentas para categorização do tráfego de rede por meio do fluxo no reconhecimento de anomalias, seja, *offline* ou *online* e vulnerabilidade *Zero Day*.

Referências

Chigada, J. and Madzinga, R. (2021). Cyberattacks and threats during covid-19: A systematic literature review. *South African Journal of Information Management*, 23(1):1–11.

Janarthanan, T. and Zargari, S. (2017). Feature selection in unsw-nb15 and kddcup'99 datasets. In *2017 IEEE 26th international symposium on industrial electronics (ISIE)*, pages 1881–1886. IEEE.

Jing, D. and Chen, H.-B. (2019). Svm based network intrusion detection for the unsw-nb15 dataset. In *2019 IEEE 13th international conference on ASIC (ASICON)*, pages 1–4. IEEE.

Moustafa, N. and Slay, J. (2015a). The significant features of the unsw-nb15 and the kdd99 data sets for network intrusion detection systems. In *2015 4th international workshop on building analysis datasets and gathering experience returns for security (BADGERS)*, pages 25–31. IEEE.

Moustafa, N. and Slay, J. (2015b). 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.

Moustafa, N. and Slay, J. (2016). The evaluation of network anomaly detection systems: Statistical analysis of the unsw-nb15 data set and the comparison with the kdd99 data set. *Information Security Journal: A Global Perspective*, 25(1-3):18–31.

Pranggono, B. and Arabo, A. (2021). Covid-19 pandemic cybersecurity issues. *Internet Technology Letters*, 4(2):e247.

Sarhan, M., Layeghy, S., Moustafa, N., and Portmann, M. (2020). Netflow datasets for machine learning-based network intrusion detection systems. In *Big Data Technologies and Applications*, pages 117–135. Springer.