Error Prevalence in NIDS datasets: A Case Study on CIC-IDS-2017 and CSE-CIC-IDS-2018

Lisa Liu[1*], Gints Engelen[2*], Timothy Lynar[1], Daryl Essam[1], Wouter Joosen[2]
[1] University of New South Wales, Canberra
{l.liuthorrold, t.lynar, d.essam}@adfa.edu.au
[2] imec-DistriNet, KU Leuven
{gints.engelen, wouter.joosen}@kuleuven.be

2022 IEEE Conference on Communications and Network Security (CNS) | 978-1-6654-6255-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/CNS56114.2022.9947235

***Abstract*—Benchmark datasets are heavily depended upon by the research community to validate theoretical findings and track progression in the state-of-the-art. NIDS dataset creation presents numerous challenges on account of the volume, heterogeneity, and complexity of network traffic, making the process labor intensive, and thus, prone to error. This paper provides a critical review of CIC-IDS-2017 and CIC-CSE-IDS-2018, datasets which have seen extensive usage in the NIDS literature, and are currently considered primary benchmarking datasets for NIDS. We report a large number of previously undocumented errors throughout the dataset creation lifecycle, including in attack orchestration, feature generation, documentation, and labeling.**

The errors destabilize the results and challenge the findings of numerous publications that have relied on it as a benchmark. We demonstrate the implications of these errors through several experiments. We provide comprehensive documentation to summarize the discovery of these issues, as well as a fully-recreated dataset, with labeling logic that has been reverse-engineered, corrected, and made publicly available for the first time. We demonstrate the implications of dataset errors through a series of experiments. The findings serve to remind the research community of common pitfalls with dataset creation processes, and of the need to be vigilant when adopting new datasets. Lastly, we strongly recommend the release of labeling logic for any dataset released, to ensure full transparency.

***Index Terms*—network intrusion detection, datasets, CIC-IDS-2017, CSE-CIC-IDS-2018.**

I. Introduction

Network Intrusion Detection systems (NIDS) form an integral part of the cybersecurity posture of any entity looking to safeguard their assets. Given the increasing complexity, heterogeneity, and volume of network traffic, it is not surprising that there is and has been great interest in using machine and deep learning models as tools that might inform cybersecurity threats. A critical component of assessing the models developed is evaluating their performance on datasets, which should reflect real-world conditions, in order to provide a realistic indicator of predictive capability. Datasets reflecting authentic network traffic in the wild cannot be gathered as they can in other domains, where datasets could be assembled from a variety of open sources. Organizations are often reluctant to release network traffic datasets because of the sensitive and confidential data they carry, whether explicit in the traffic payload, or implicit in the connection meta-data. To navigate the issue of inaccessibility to authentic traffic datasets, researchers have turned to creating their own datasets as a proxy to real-world network traffic using controlled test environments with synthesized attacks.

Among the first datasets provided to the research community were the DARPA [1] and it's derivative, KDD Cup '99 [2] datasets, which served as long-standing benchmarks for NIDS research. Both contributions allowed for the first time meaningful comparison of different approaches and advanced the field significantly. However neither datasets have been immune to criticism [3], [4]. The criticisms centered on the presence of simulation artifacts, redundant records, data corruptions, misalignment to production network architectures, and with passage of time, lack of contemporary relevance. NSL-KDD [5], an improved iteration of KDD, resolved the redundant record issue, and became the successive mainstay benchmark dataset for NIDS research, alongside UNSW-NB15 [6], CICIDS-2017 [7], CSE-CIC-IDS-2018 [8], and BoT-IoT [9]. These datasets have been released as post-processed forms of raw network traces, and have been labeled for easy pairing with machine learning algorithms. The authors justify the contributions of these datasets by emphasizing the need for contemporary attack scenarios, larger network configurations, and a richer feature set for machine learning algorithms. The primary form of validation for these datasets comes from their demonstrated ability to differentiate between malicious and benign flows using machine learning algorithms.

With these datasets providing a large number of records - for example, UNSW-NB15 and CICIDS-2017 contain close to 2,000,000 records, and CSE-CIC-IDS-2018 has an excess of over 16,000,000 records - there is no alternative but to apply an automated labelling process. This is usually done through a combination of time and host-based filtering to separate malicious traffic [10]. Formulating the logic is a key component of label generation, a manual and laborious process, due to the quantity and complexity of data involved, making it susceptible to errors if not handled carefully. The validity of the labeling (ground-truth) has, to our best knowledge, not been established.

The problem has adverse implications for the field. The number of datasets for benchmarking Network Intrusion Detection Systems (NIDS) has proliferated dramatically to meet the current conditions of local and global network traffic, which

*Co-first authors.

2022 IEEE Conference on Communications and Network Security (CNS)

has undergone drastic changes due to sweeping technological innovations in recent years. It is often the case that these datasets are released without an intensive external validation process, and are then directly adopted by users, resulting in misleading claims of extrapolated efficacy in the case that datasets are found defective.

With the number of issues that are discovered in the CICIDS-2017 dataset [11], [12], the extent to which CSE-CIC-IDS 2018 has been compromised by the same issues, as well as other undiscovered problems, will be of interest, given the lineage to CIC-IDS 2017. Both datasets have served as the primary benchmark in many recent published studies [13]–[21]. A pairing of CIC-IDS 2017 and CSE-CIC-IDS 2018 also provides a unique opportunity to assess generalization capabilities of developed models based on similar dataset constructions in benign and attack traffic [22], [23]. The two datasets are distinguished by their testbed configuration and deployment environment, which serve to represent real-world scenarios where datasets may be trained in one environment and deployed to another.

We perform a rigorous methodological manual analysis to determine the extent to which both datasets are affected by labeling issues. To our best knowledge, our work represents the first to attempt to reverse-engineer the closed-source labeling logic for these datasets from raw .pcap files. In summary, the contributions of the work are as follows:

- We identify a large number of previously undocumented errors throughout the CIC-IDS 2017 and CSE-CIC-IDS 2018 dataset creation lifecycle, including in attack orchestration, feature generation, documentation, and labeling.
- We provide completely reconstructed datasets that are free of artifacts and use a corrected labeling logic that has been engineered by manually analysing over 630 GB of raw traffic and 31.14 GB of CSV flow data.
- The new datasets include corrected implementations of existing features, and provide new features that capture valuable flow state information, which is not provided by the current feature extraction tool.
- We demonstrate the effect of dataset errors through a series of experiments.

II. Background and Related Work

A. Datasets

The CSE-CIC-IDS 2018 dataset is published by the University of New Brunswick through a collaboration between the Communications Security Establishment (CSE) and the Canadian Institute of Cybersecurity (CIC). The dataset follows the CICIDS2017, ISCXIDS2012, and NSL-KDD datasets released by the same research institution, all of which have been extensively used for NIDS benchmarking. Each iteration expanded the capabilities of the previous version through larger network testbeds, or new attack scenarios. CSE-CIC-IDS 2018 was largely unchanged from the 2017 iteration, with the exception of moving the testbed to Amazon Web Services cloud infrastructure and increasing the number of simulated clients in subnetworks. The attacks contained within the 2017 iteration are mostly retained in the 2018 version.

The dataset itself consists of 10 days of traffic, presented in .csv format, with 79 features extracted through the CICFlowMeter Extraction tool [24], [25]. The flows are bi-directional, defined by the 5-tuple {*Source IP, Destination IP, Source Port, Destination Port, Protocol*}, with the first packet in the flow defining the forward direction. A flow is terminated by a user specified timeout, or by TCP connection semantics. The main flow features include values (time, data, packet rates), their statistics (min, max, mean, standard deviation, variance), and indicator variables for TCP header flags. The labeling is categorized by attack type, enabling both multi-class and binary classification. Benign traffic is produced from B-Profiles which extract human behavior into encapsulated features. For example, distribution of packet sizes of a protocol, number of packets per flow, and certain patterns in the payload. According to the documentation [26], [27], B-Profiles are used by an in-house, closed-source Java application to generate realistic network traffic in the backdrop of traffic noise over the following protocols: HTTPS, HTTP, SMTP, POP3, IMAP, SSH, and FTP. There are no further details about how benign traffic is generated. For example, the sampling duration, the network environment of profiled users, or statistical details of simulated behavior are not readily apparent. The attack scenarios are described in their documentation [26], [27].

B. Related Work

NIDS datasets have often been analyzed in the literature. A comprehensive survey of datasets relevant to network-based intrusion detection is presented by [28]. Despite the review's good coverage of the many datasets available, the comparisons rely on published documentation to illustrate differences between datasets, which themselves are limited in accuracy to what is contained therein. In [29], the authors conducted a similar study, and raised cursory issues such as the size of the dataset, missing or duplicate records, and class-imbalance.

A recent review of studies involving CSE-CIC-IDS 2018 up to 2020 can be found in [30]. We draw attention to several studies in the review where the levels of accuracy, precision, and recall have reached 100%. As we later demonstrate, it is highly unlikely that the stated performance can be achieved without overfitting to the dataset, given the level of label contamination we observe. The inability to extrapolate conclusions beyond the performance of the datasets themselves reveals limited insight into real-world performance. [31] reached a similar conclusion, based on the observed biased nature of the datasets. In their statistical analysis of CIC-IDS 2017 and UNSW-NB15, among others, they found that a number of the features were over-correlated with anomalous classes and that the probability distributions between attacks and normal traffic differed substantially. They conclude that these datasets should not be used as a benchmark for creating novel anomaly-based network intrusion detection systems, given that these features alone were deemed sufficient to make a distinction between benign and malicious traffic.

III. Methodology

CSE-CIC-IDS 2018 is published without certain features which enable reconciliation of .csv flow features with raw packet traces from .pcap files. These include: Flow-IDs, Source/Destination IPs, and Source Port Numbers. Following our contact, the dataset authors provided the complete .csv files with the missing fields, but were unable to supply the scripts used to label the flows in either dataset. The authors in this study subsequently use the documentation located at [26], [27] as the primary reference points for reverse engineering the correct labeling logic for both datasets. The study was conducted by two authors over a period of 5 months, during which each attack was manually examined in great detail. Both authors have previously analyzed a number of NIDS datasets, including raw .pcap files over the past 2.5 years.

Each author independently performed the manual analysis. The scope of work included locating evidence of each attack in the raw .pcap files, identifying labeling inconsistencies in the published versions with reference to packet traces, and examining the published .csv files, containing extracted flows from the CICFlowMeter tool [25], for accuracy in feature extraction. Independent findings were mutually verified before coming to a consensus on the final labeling scripts, with multiple rounds of cross-validation during each stage of the study. The new labeling logic was then applied to the original dataset. Inconsistencies between the new ground truth and the original published labels were manually verified. Following modifications to the CICFlowMeter tool to incorporate corrections to faulty implementation of existing features, addition of stateful features, removal of artifacts, the new datasets were regenerated. After application of the corrected labeling logic to the regenerated datasets, the new datasets were re-evaluated following the same procedure. The methodology was used to analyse both CIC-IDS 2017 and CSE-CIC-IDS 2018 datasets.

IV. Evaluation Results

In this section, we first provide a quantitative summary of issues found within each dataset. A summary of modifications made to the CICFlowMeter tool is then provided. This is followed by experiments examining their impact on learning behavior in machine and deep learning algorithms. Finally, the results of an inquiry into the automated detection of labeling errors are presented.

A. Quantitative Summary of Issues

The quantitative assessment covers attack label omissions, significant label reassignments, and the issue of ambiguous class labels (class overlap). Our analysis is based on the published version of CIC-IDS 2017, and on the *full* version of CSE-CIC-IDS 2018 (which includes the omitted columns of Flow-ID, Source/Destination IPs and Source Port), obtained after correspondence with the authors. We discover that the published version of CSE-CIC-IDS 2018 is only an approximate 25% subset (16,232,963 flows) of the *full* version. We remove rows containing invalid numeric values. The calculations hereafter are based only on the flow counts of rows with valid numerical values (cleaned flow count). Statistics of the analyzed datasets are presented in Table I.

TABLE I
Dataset Statistics

Year	Published Flows Count	Cleaned Flows Count	CSV Volume GB	PCAP Traffic GB
2017	2,830,743	2,827,726	1.09 GB	48.8 GB
2018	65,042,705	64,561,010	30.05 GB	582 GB

1) Attacks Missed By Published Version: Table II summarizes the number of malicious flows that are missed in the original dataset. The issue is important, since the dataset is severely imbalanced in favor of benign traffic. Flow count gains are especially significant for attacks that are challenged by a minority representation in the dataset. Imprecise time frames account for some of the missed flows. Some attacks were missed entirely, e.g. several rounds of Port Scanning (2017, 2018). Some labels were assigned to the incorrect attack. Substantial gains are accounted for by considering several issues which lead to an unexpected direction in flow initiation, see Section IV-B. In the published dataset versions, the labels are mostly considered in one direction only.

TABLE II
Attack Labels Missed In Published Version

	2017		2018	
Attack	Count*	%Gain*	Count*	%Gain*
DDoS LOIC	[128,025] +31,339	24.48%	-	-
DDoS HOIC	-	-	[1,246,034] +918,543	73.72%
DoS GoldenEye	[10,293] +203	1.97%	-	-
DoS Hulk	[230,124] +3,680	1.60%	[923,824] +884,408	95.73%
DoS Slow HTTP	[5,499] +163	2.96%	-	-
DoS Slowloris	[5796] +10	0.17%	-	-
FTP Patator	[7,935] +19	0.24%	-	-
Heartbleed	[11] +1	9.09%	-	-
Infiltration	[36] +4	11.11%	[160,639] +63,381	39.46%
Port Scan	[158,804] +61,003	38.41%	-	-
SSH Patator	[5,897] +1	0.02%	[187,589] +389	0.21%
Web Brute Force	[1,507] +1	0.07%	-	-
Web SQL Injection	[21] +3	14.29%	[87] +1	1.15%
Web XSS	[652] +2	0.31%	[230] +2	0.87%

* Values with reference to label counts in the published dataset.

2) Significant Label Reassignments: The discovery of labelling issues in the original datasets leads to a number of label reassignments. Table III presents the instances where corruption to the published label exceeds 5% of the label's total flow count. Extremely high corruption rates can be observed for many attacks, in both datasets. The table does not represent the entirety of all label reassignments as many fall below the 5% corruption rate. We provide below a brief summary of main factors leading to label re-assignment.

- **Empty Payload** The primary contributions to this category are "TCP Appendices" [11]. There are also flows with only TCP start and finish sequences and no traffic in between.
- **Port/System closed** Malicious traffic does not reach victim target due to it being down or unavailable.

TABLE III
LABEL REASSIGNMENTS: CORRUPTIONS > 5%

Year	Published Label	Revised Label	% Corruption*	Remarks*
2017	Bot	Botnet - Attempted	25.05%	Port/System Closed. Continued C&C connection attempts by victim after attack terminated at the published time, and C&C is no longer reachable (presumably shut down).
	DoS GoldenEye	DoS GoldenEye - Attempted	25.06%	Empty Payload
	DoS Hulk	DoS Hulk - Attempted	32.70%	No malicious payload (4.47%) Empty Payload (28.23%) Attack artifacts (0.001%)
	DoS Slow HTTP	DoS Slow HTTP - Attempted	56.34%	Empty Payload (5.13%) Target Unresponsive (51.19%) Attack startup/ tear down artifact (0.02%)
	DoS Slowloris	DoS Slowloris - Attempted	29.65%	Empty Payload (29.55%) Attack startup/ tear down artifact (0.10%)
	FTP-Patator	FTP-Patator - Attempted	49.93%	No malicious payload (49.69%) Empty Payload (0.20%) Attack startup/ tear down artifact (0.04%)
	SSH-Patator	SSH-Patator - Attempted	49.80%	No malicious payload
	Web - Brute Force	Web - Brute Force - Attempted	95.16%	Empty Payload (89.85%) Attack startup/ teardown artifact (0.60%) Attack artifact (4.71%)
	Web Attack - XSS	Web Attack - XSS - Attempted	94.48%	Attack startup/tear down artifact (0.61%) Empty Payload (93.87%)
2018	Bot	Botnet Ares - Attempted	50.06%	Empty Payload. No evidence of Botnet Zeus found during traffic analysis. Affected by TSO issue.
	Web - Brute Force	Web - Brute Force - Attempted	53.85%	Empty Payload (30.77%) Attack startup/ tear down artifact (2.46%) Attack Implemented Incorrectly (20.62%) - Flows with only single login requests do not qualify as 'brute force' in the context of a single flow
	Web - Brute Force	Benign	24.71%	Mixture of empty flows comprising TCP handshake only, and browsing traffic not specifically related to the attack.
	Web Attack - XSS	Web Attack - XSS - Attempted	50.43%	Attack startup/tear down artifact (1.30%) Empty Payload (48.26%) No malicious payload (0.87%)
	DDoS HOIC	DDoS HOIC - Attempted	86.86%	Empty Payload
	DDoS LOIC-HTTP	DDoS LOIC-UDP	0.14% (LOIC) 46.07% (UDP)	All DDoS LOIC-UDP flows on 20-02-2018 are mislabeled as DDoS LOIC-HTTP
	DDoS LOIC-HTTP	DDoS-LOIC-HTTP - Attempted	49.65%	Empty Payload
	DoS GoldenEye	DoS GoldenEye - Attempted	33.22%	Empty Payload (33.09%) Target Unresponsive (0.13%)
	DoS Hulk	DoS Hulk - Attempted	94.47%	Empty Payload
	DoS Slow HTTP	FTP-Patator - Attempted	100%	Attack mislabeled. Traffic shows TCP connection attempts to port 21 which appears to be closed
	DoS Slowloris	DoS Slowloris - Attempted	21.88%	Empty Payload
	FTP-Patator	FTP-Patator - Attempted	100%	Port/System Closed. Unsure if this was an accidental oversight. All TCP 'Syn' connection attempts to victim host/port are responded to with 'RST'.
	Infiltration	Benign	76.84%	Hosts not involved in the attack are classified as infiltration. Suspect severe contamination of background traffic due to time-based labelling
	SQL Injection	SQL Injection - Attempted	54.02%	Empty Payload (36.78%). Some flows in this category occur in the timeframe of another attack. Attack startup/tear down artifact (17.24%)
	SSH-Patator	SSH-Patator - Attempted	49.97%	Empty Payload

* Values accurate at time of publication. For the full table (including corruptions < 5%), refer to `https://intrusion-detection.distrinet-research.be/CNS2022/`. There may be minor residual corrections post-publication.

- **Attack startup/teardown artifact** Traffic with no malicious semantics. They are related to launching or terminating the attack. For example in FTP Patator (2017), the traffic suggests a successful trial login takes place before the start of the attack. It is possible that the dataset author is verifying that the victim is configured correctly for the attack. Most web-based attacks require a first visit to the landing page of DVWA (vulnerable web-app) installed on the victim server, before proceeding to the appropriate page to load the attack. Due to the absence of malicious payload in this phase, they appear semantically identical to a benign user browsing the web-app, within the context of a single flow.
- **No malicious payload** Payload exists, but without malicious content. For example, some long TCP connections are split into multiple flows due to exceeding the timeout of the CICFlowMeter tool (set at 120 seconds). These partitioned flows created by flow timeouts do not always contain malicious traffic. Other flow types in this category include re-transmission of small segments without malicious components, and flows with RST packets that are technically affiliated TCP appendices.
- **Attack artifact** Traffic occurring between attacker and victim, but unrelated to the attack. For example, during the Web-Brute Force attack (2017), a HTTP GET request to the page `/dv/vulnerabilties/xss_r/` is made with every HTTP POST login request to `/dv/login.php`. The consecutive HTTP requests are split into separate flows, the former with no malicious brute force content. Our manual replication of the login attempt to the specified page with our own setup of DVWA does not yield the same traffic patterns; the unrelated HTTP GET request may be an introduced programmatic artifact during attack execution.
- **Target system unresponsive** The target is unresponsive, suggesting that it may be overwhelmed. This is distinguished from Port/System closed, where a RST packet is returned to the sender. Flows in this category consist primarily of TCP 'Syn' packets submitted to the victim host through TCP re-tranmission mechanisms.
- **Time-Based labeling** Without accurate host and port filtering, time-based labeling leads to traffic not involved in the attack being labeled as malicious flows. This gave rise to high levels of contamination to Infiltration (2018), and Web-Brute Force (2018).

3) Ambiguous Class Label: We identify a number of flows which, once flow identifying features [Flow Id, Source IP, Source Port, Destination IP, Timestamp] are removed, share overlap with other classes, presented in Table IV. This has several consequences. First, the presence of TCP appendix artifacts is a systematic issue that affects all TCP flows. Artifacts of this type are found in almost all classes, and share similar characteristics. The second is a result of mislabelling. Recall from Table IV that the DoS SlowHTTPtest attack is entirely mislabeled. The raw packet trace analysis from .pcap files during this period reveals FTP connection attempts to an inactive host/port. Third, due to a number of incorrectly implemented columns in the CICFlowMeter tool used at the time the datasets were generated, additional features that could provide flow disambiguation between classes fail to be expressed accurately, if at all, in the affected fields.

TABLE IV
AMBIGUOUS LABEL COUNTS

Year	Attack	Overlap Class	# Overlap
2017	DDoS	Benign	6
	DoS Hulk	Benign	4,872
	DoS Slowloris	Benign	1
	Portscan	Benign	1,053
2018	Brute Force - Web	Benign	1
	Web - Brute Force	Infiltration	7
	Web Attack - XSS	Benign	1
	Web Attack - XSS	SQL Injection	1
	DoS Slow HTTP	FTP-Patator	100,760
	DoS Slow HTTP	SSH-Patator	89,438
	FTP-Patator	SSH-Patator	169,745
	Infiltration	Benign	36,889

B. Modifications to CICFlowMeter Tool

In this section, we summarize key issues/changes with the CICFlowMeter that are pertinent to flow integrity and attack characterization. Corrections where possible have been made[1].

- **Packet timestamping issue** As an example, in FTP-Patator (2017), we observed some 'Syn Ack' packets from the victim arrive (in the temporally-ordered trace) before the initiating 'Syn' packets from the attacker. The result is an unexpected initiation of flows from the victim. Local network interface cards (NICS), which are then combined into one logical NIC, are likely to cause this issue, since timestamping occurs on the OS where the local NICS reside. This issue is likely to surface in volumetric attacks. To remedy this, a dataset creator should verify both directions of traffic when labeling flows.
- **TCP segmentation offset issue** A valid IP packet from an interface performing TCP segmentation offloading (TSO) may have an IP length of 0 in the header. Since packet headers currently determine how packets are dissected, affected packets are dissected incorrectly. As a result, packets that would usually be assigned to the correct flow, are instead separated out into their own flow with source and destination ports set to 0 ('unknown payload'). Multiple flows between the same set of hosts, with different ports, occuring in the same timeframe, will also have any TSO packets collected into the same 'unknown payload' flow. This is likely to affect attacks with large payloads.

[1]The corrected CICFlowMeter tool can be found at `https://github.com/GintsEngelen/CICFlowMeter`

Observed in Heartbleed (2018) and Botnet Ares (2018). Our release contains a modification to the `jnetpcap` library used by the CICFlowMeter to address this issue.

- **Correct calculations of existing attributes** Issues in this category were due to erroneous implementation in the original version of the CICFlowMeter tool. The affected features are therefore inaccurate representations of reality.
- **New attributes**
  1) *Add total TCP flow duration.* A user can now identify whether a CIC Flow belongs to a sustained TCP connection. The use of lengthened TCP connections is particularly relevant to low rate DoS attacks that are designed to exhaust a victim's connection pool.
  2) *Support for ICMP protocol*. The ICMP protocol is not supported by the original CICFlowMeter. Undelivered IP packets may be embedded and returned to the sender through ICMP, which is then incorrectly interpreted. A regular flow is subsequently created using the Source and Destination IP addresses of the ICMP packet, and the protocol, port, and payload information in the encapsulated, undelivered packet.
  3) *Fwd/Bwd RST flag counts introduced*. Anomalous activity may be indicated by direction of RST flags.
  4) *UTC timestamp with micro-second precision*. The timestamp is standardized to UTC, rather than to the local host running the Flow Meter. This removes ambiguity when datasets are analyzed from different time zones. With microsecond precision, it is possible to identify the exact packet that initiated a flow in the raw .pcap files. Precise packet reconciliation facilitates attribution to unexpected observations.

C. Impact on Training

In this section we study the cumulative impact of issues present in the CIC-IDS 2017 and CIC-CSE-IDS 2018 datasets. We evaluate the impact of labeling errors on standard ML/DL algorithms, since judgment of validity cannot be cast on existing state-of-the-art-methods. As we discuss later below, existing methods that perform well on the published versions of the datasets, with the levels of labeling contamination observed in this study, are likely demonstrating their ability to fit to the chosen dataset, rather than ability to learn appropriate representations to solve the problem at hand. First, we discuss differences in feature representation on Machine Learning algorithms, followed by the impact on training of Deep Learning Algorithms.

1) Differences in Feature Representation: To analyze differences in feature representation of attacks between the published and newly generated datasets, we select Random Forest, a ML classifier which consistently stands out as a high-performing algorithm in NIDS research [32]–[37]. From the 2017 and 2018 dataset versions, we select 2 attacks each with high corruption rates. The analysis procedure is as follows: For all datasets, we first drop flow-identifying columns. We also drop columns that are strongly correlated with others, determined by using Pearson correlation, with a coefficient of 0.9 as the threshold. We then calculate the baseline performance for each version of the dataset. Afterwards, we systematically remove one feature, retrain the classifier, and recompute the Precision, Recall, and F1 scores. We repeat this for all features.

We interpret a feature's importance by the impact it has on Precision, Recall, and F1 values to the baseline metrics when that feature is removed. Figure 1 illustrates the reduction in F1 scores when each feature is removed. The selected features in the figure represent the top *five* features for each class, from each version of the dataset (published/released), giving a total of *ten* features for comparison in each attack. Baseline Recall and F1 results (using the full set of decorrelated columns) for each attack are presented to the right of the subplots. In observing impact to F1 scores, both in terms of feature types and contribution level, one can view differences in *learning representations* when dataset artifacts and labeling corruptions are present. These differences result in elevated Recall and F1 performance for nearly all cases in the corrected version.

It is possible to see differences in learning representation even when the metrics for a given attack are quite comparable in both dataset versions, e.g. 2017 DoS Slow HTTP. Typically, *Destination Port* and *Forward Initial Window Bytes* are prominent features for many attacks in the published versions of the datasets, including the DoS Slow HTTP. We establish a link to these features through the presence of artifacts, which outweigh the quantity of genuine malicious flows. For many attacks, the importance of *Destination Port* and *Forward Initial Window Bytes* strongly diminishes in the corrected dataset versions.

Several other interesting observations can be made. Firstly, we observe some attributes have a negative impact on classification performance (i.e. drop in F1 score when the feature is removed). Due to the overlap in distributions of values to other classes, not all features lead to quality predictive indicators for the target class. This is likely to have occurred in DoS Slowloris (2018, published version) due to mislabeling or the presence of artifacts within the values for the affected features. The second observation is that singular feature importance drops substantially in the corrected versions of the datasets. In other words, dropping single features in an independent run resulted in no change to model performance for most of the features, and for many attacks. On one hand, this can be interpreted as an encouraging sign, since strong attachment to singular features may be symptomatic of shortcut learning. When a variable is dropped without changing the model, it indicates that the information needed for classification can be provided by one or more of the remaining variables. We consider two scenarios: 1. Multi-collinearity between other features 2. Extremely low intra-class variability for many of the attacks, due to their scripted nature. This leads to more or less consistent values *within* features (for several features) for a given attack. In light of our empirical observations of the datasets in purified form, we hypothesize that (2) is the most likely explanation.

2) Impact on Deep Learning Training: To gain insight into how dataset issues impact Deep Learning algorithms, we study

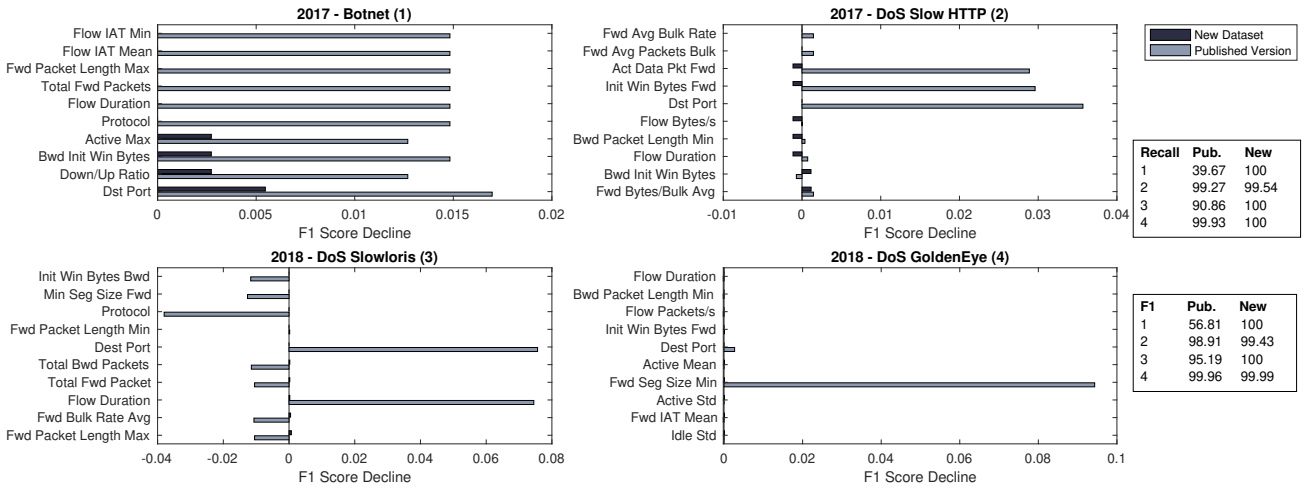

Fig. 1. Learning Representation Differences - Random Forest

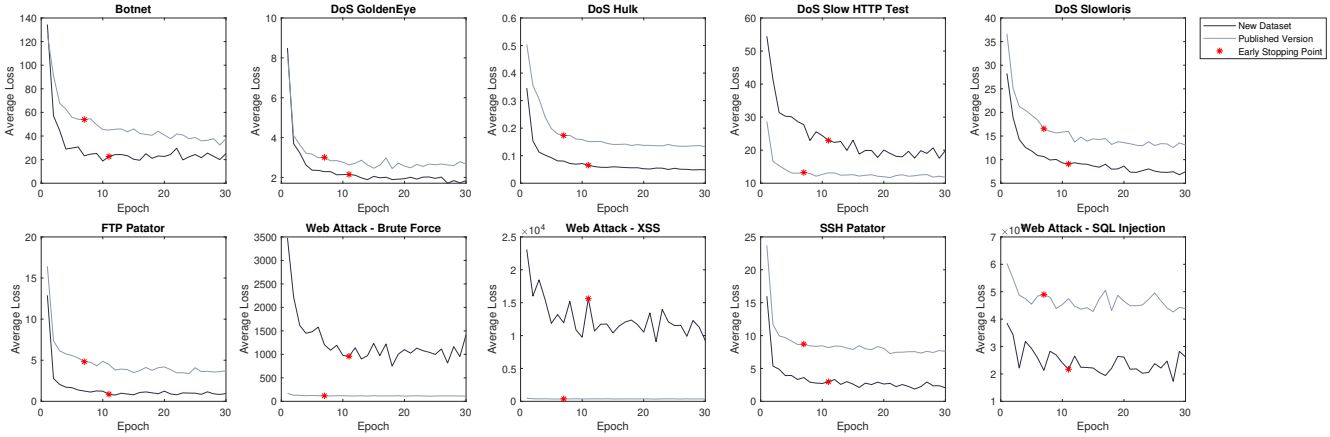

Fig. 2. CIC-IDS2017 Average Categorical Cross-Entropy Losses vs. Epochs Comparison

and compare the training losses for each attack between the published versions and our corrected versions. The intuition is that the learning behavior of a class can be reflected through their training losses [38]. The configuration of a Multilayer Perceptron was derived through hyper-parameter optimization, using the Hyperband algorithm [39]. The final architecture is a standard MLP with 4 hidden layers. For each version of the dataset, we track the sample-wise training loss for each training epoch, up to a total of 30 epochs. The early stopping point for models trained on each version of the dataset is marked in red on the corresponding subplots of Figures 2 and 3. The early stopping point is determined by an increase in validation loss, with a patience of 3 epochs. The categorical cross entropy losses are weighted according to class sizes.

Figures 2 and 3 illustrate the individual sample losses, aggregated by attacks for each version of the dataset (2017 and 2018, respectively). In general, we can observe lower training losses for the corrected dataset versions in nearly all attacks, indicating increased stability in training, with corrected labeling and removal of artifacts. In the cases where training losses in the corrected version of the dataset exceed the losses in the published version, most are impacted by their minority representation in the main datasets. The removal of artifacts and incorrectly labeled flows further reduces their flow count availability, which increases the training difficulty. In these scenarios, they correctly reflect an increased training difficulty on minority classes, without artifact contamination.

For the DoS Slow HTTP Test (2017) example, we observe a higher average training loss in the corrected version of the dataset, compared with the published version. In our investigation, we found that the training loss progression of the published version of DoS Slow HTTP aligns closer to the training loss progression of 'DoS Slow HTTP - Attempted' in the corrected version. This suggests that the learning representation of the DoS Slow HTTP Test may be tied to TCP appendices in the MLP trained on published versions of the dataset. Once these are removed, the training losses are calibrated to the true learning difficulty of this attack.

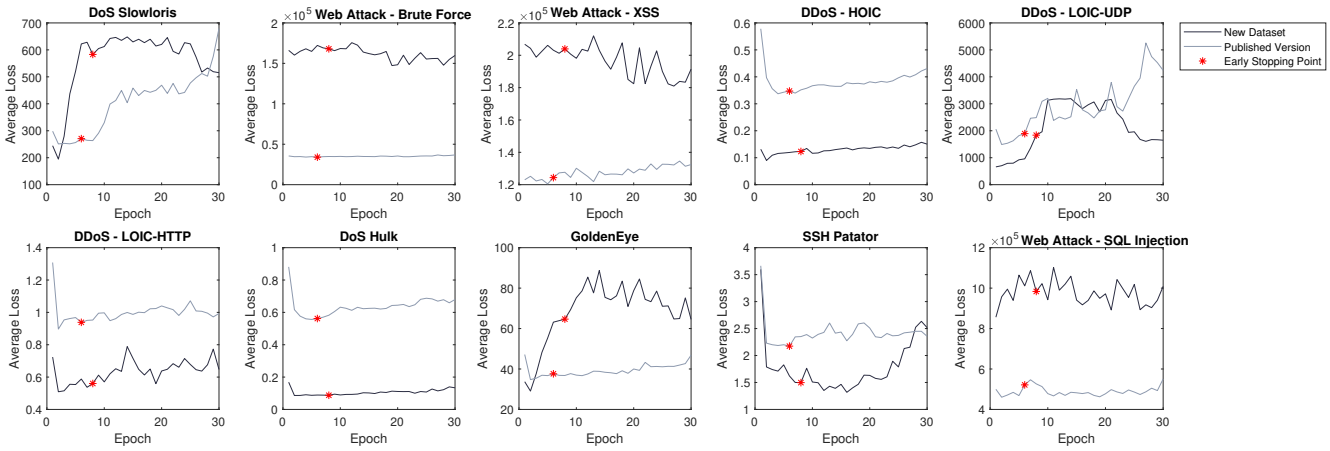

Fig. 3. CSE-CIC-IDS2018 Average Categorical Cross-Entropy Losses vs. Epochs Comparison

D. Automated Detection of Labelling Errors

Manual reverse engineering and correction of labeling logic provide us with a ground-truth reference point from which to investigate the possibility of automated detection of labeling errors in the two datasets. We select two state of the art methods, Confident Learning [40], and O2U-Net [38] that have been used to detect labeling errors in the field of computing vision. As the base algorithm for Confident Learning, we use Random Forest. The MLP configuration from the previous subsection is used as base model for O2U-Net. The *ground-truth* for comparison is generated by applying our corrected labeling logic to the *published* datasets (in the case of CIC-CSE-IDS 2018, the *full* version).

In Table V, the mismatch counts to ground-truth are shown for the published versions, as well as the labels as predicted by Confident Learning and O2U-net. The baseline label corruption levels (i.e. the published versions of the dataset) sit at 6.67% and 7.53% for 2017 and 2018, respectively. The mismatch counts for Confident Learning are comparable to the baseline corruption levels, with slight decreases for both 2017 and 2018. The mismatch counts are considerably high for O2U-Net in both cases. We speculate that in addition to the presence of the artifacts, deep learning algorithms are less robust to dealing with severe data imbalances. This is despite using hyper-parameter optimzed networks, with weighted losses. On the contrary, as earlier discussed, there is a difference in learning representations between a clean dataset, and one in which artifacts are present. It is possible that exposure to artifacts in the Random Forest classifier contributes to its tendency to present mismatch counts in close proximity to the published versions. This suggests that Confident Learning can detect only spurious labeling errors.

While both methods are model agnostic, their performances are reliant upon the underlying algorithm's ability to extract optimal semantic representations. We conclude that both methods are challenged by the presence of dataset artifacts and are unable to offer substantial improvements to the correction of labeling errors in CIC-IDS 2017 and CSE-CIC-IDS 2018.

TABLE V
AUTOMATED LABEL ERROR DETECTION

	2017		2018	
Total Flows	2,827,726		64,561,010	
Incorrectly labeled	**Count**	**% Mismatch**	**Count**	**%Mismatch**
Published Version (Baseline)	188,571	6.67%	4,862,181	7.53%
Confident Learning [40]	188,489	6.67%	4,781,570	7.40%
O2U-Net [38]	412,330	14.58%	12,655,025	19.60%

V. DISCUSSION

Implications of Ambiguous Label Counts. It is not uncommon to find datasets with ambiguous labels. Usually, they arise from subjectivity in human interpretation for contentious samples. This is especially the case for the field of computing vision. It is more unusual to see post-processed samples from different traffic scenarios presenting identical values for every field in the extracted flows. In these datasets, the findings suggest that the ambiguous labels are caused by a mixture of artifact presence and labeling inaccuracies. In general however, this is symptomatic of an underlying feature representation problem. Feature extraction techniques that produce identical features with multi-class assignments, that should not exist, implies that the engineered features are insufficient for effective training with Machine and Deep Learning algorithms. The effects in our study show some degree of training destabilization. However, when samples with class overlap surface in the testing/validation partitions of the dataset, this makes theoretical 100% accuracy impossible, since a single sample cannot be simultaneously several labels in a single class prediction problem. It is possible to avoid the latter problem by removing duplicate rows before commencing the ML pipeline, but it does not address the underlying issue of a feature representation problem. There is no guarantee that the behavior of learning algorithms on unseen samples will be predictable.

Implications on Learning Representations. The automated process of labeling records raises the possibility that machine learning algorithms learn features that reflect the logic of the labeling process rather than relevant characteristics that

delineate benign from malicious records. Recent studies have demonstrated that models may overfit to patterns in the data, including patterns associated with systematic labeling errors [41], or even a random labeling of data [42]. Both forms of learning are capable of delivering high levels of performance. Learning the former provides an unrealistic estimation of the model's performance in the field, as well as being detrimental to any generalization capability the model is able to provide. Practitioners will not be able to differentiate between the two cases without detailed information on the labeling strategy. This issue is further obscured by the volumetric and imbalanced nature of these datasets. When feature reduction, a standard technique in the ML pipeline is used, these problems are exacerbated further because the algorithms may focus more disproportionately on irrelevant features.

A further consequence of incorrect learning representations caused by dataset issues, is erroneous conclusions in extrapolated efficacy, as well as incorrect interpretation of technique performance. As an example, the Infiltration attack in CSE-CIC-IDS-2018 suffers from severely high levels of label corruption, an attack that has been identified as notoriously difficult to detect [32]. A researcher may be led to optimise toward perceived algorithmic deficiencies rather than recognise that the level of contamination makes this an impossible task. In view of these findings, we believe that new metrics must be developed to evaluate the performance of ML/DL NIDS research, since methods that benchmark well, the current qualification for state-of-the-art, may be invalid by virtue of learning inappropriate representations to solve the problem at hand. For example, measures of generalization capability may offer a more accurate representation of real-world performance, allowing us to disqualify methods that perform well on benchmark datasets only.

Implications on NIDS research. NIDS dataset paper releases have kept labeling logic closed source as standard practice. This is problematic for several reasons. A user cannot easily verify the simulated environment has been adequately controlled for the artifacts it generates. Among many others, these include background traffic between attack and victim machines during attacks, simulation tool artifacts, and anomalous behavioral manifestations by the machines following completion of the main attack. It is our *strong* recommendation that any future dataset releases undergo an external, independent validation process before they can be accepted as a valid benchmark dataset for NIDS research. To ensure integrity of dataset creation, the labeling logic must also be publicly accessible, as well as complete, accurate, and thorough documentation. We remark the findings of our study are confined to CICIDS-2017 and CSE-CIC-IDS-2018, the primary benchmark datasets for NIDS research in the last five years. It is unknown to what extent other datasets suffer similar problems. It is a significant undertaking to determine the correct ground-truth necessary to establish their validity as benchmark datasets.

VI. Threats to Validity

In this section, we take into consideration potential threats to the validity of this study. While our work allows for the correction of label errors and removal of artefacts at the post-processed levels, the datasets are not free of shortcomings. Alongside inherent limitations of datasets generated in synthetic environments, our work is unable to remediate issues at the raw .pcap level. For example, the misimplementation of the DoS Hulk attack renders it completely ineffective. This requires re-execution with corrected implementation tools on the original testbed architectures.

Finally, while we carefully reverse engineered the corrected logic in both datasets over the course of many months, we acknowledge that analyzing datasets of this size is extremely challenging, and may contain small residual errors. By publishing our labeling logic efforts and the accompanying documentation (over 160 pages of analysis)[2], we hope to foster future communication with the research community in order to address any future corrections. Our research team is glad to make periodic releases of datasets with added corrections identified by the research community.

VII. Conclusions and Future Work

Benchmark datasets are heavily depended upon by the research community to validate theoretical findings and track progression in the state-of-the-art. NIDS dataset creation presents numerous challenges on account of the volume, heterogeneity, and complexity of network traffic, making the process labor intensive, and thus, prone to error. In this work we conducted an in-depth manual analysis of the CIC-IDS-2017 and CSE-CIC-IDS-2018 datasets to validate their integrity. Throughout our investigation, we uncovered a number of issues pertaining to feature generation and labeling.

As a consequence of labeling inaccuracies, corruption of labels is significant for many attacks. For the entire datasets of CIC-IDS-2017 and CSE-CIC-IDS-2018, labeling errors totaled 6.67% and 7.53% respectively. Corruption rates to some attacks exceeded 75%. Remediations to these issues resulted in altered learning representations that better reflect the characteristics of attacks, leading to their improved detection rates. Increased training stability for deep learning methods was also observed. We expect that the generalization capabilities will also be strengthened based on the recalibrated learning representations. To the research community, we provide reconstructed datasets, our reverse-engineered labeling logic, as well as comprehensive documentation detailing the summary of our investigations.

Our next research priorities are focused on developing new methods for the automated detection of labeling errors in NIDS datasets. We envision this will assist in future dataset creation efforts, as well as providing a solution of label correction for datasets already published. We hope our public release of corrected datasets and their labeling logic will drive further research initiatives in this area.

[2]https://intrusion-detection.distrinet-research.be/CNS2022/

ACKNOWLEDGMENTS

Lisa Liu wishes to acknowledge the Commonwealth's contribution, for the provision of support provided through an Australian Government Research Training Program (RTP) Scholarship. Gints Engelen wishes to thank the Research Fund KU Leuven and by the Flemish Research Programme Cybersecurity for helping to fund this research.

REFERENCES

[1] "Lincoln lab datasets." https://www.ll.mit.edu/r-d/datasets. Accessed: 2022-2-6.
[2] "KDD cup 99." https://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.
[3] J. McHugh, "Testing intrusion detection systems: a critique of the 1998 and 1999 DARPA intrusion detection system evaluations as performed by lincoln laboratory," *ACM Trans. Inf. Syst. Secur.*, vol. 3, pp. 262–294, Nov. 2000.
[4] V. Engen, J. Vincent, and K. Phalp, "Exploring discrepancies in findings obtained with the KDD cup '99 data set," *Intell. Data Anal.*, vol. 15, pp. 251–276, Mar. 2011.
[5] M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, "A detailed analysis of the KDD CUP 99 data set," in *2009 IEEE Symposium on Computational Intelligence for Security and Defense Applications*, pp. 1–6, July 2009.
[6] N. Moustafa and J. Slay, "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)*, pp. 1–6, Nov. 2015.
[7] I. Sharafaldin, A. H. Lashkari, and A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," *ICISSp*, vol. 1, pp. 108–116, 2018.
[8] "CSE-CIC-IDS2018 2018 dataset." https://registry.opendata.aws/cse-cic-ids2018/.
[9] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, "Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset," *Future Gener. Comput. Syst.*, vol. 100, pp. 779–796, Nov. 2019.
[10] J. Guerra, C. Catania, and E. Veas, "Datasets are not enough: Challenges in labeling network traffic," Oct. 2021.
[11] G. Engelen, V. Rimmer, and W. Joosen, "Troubleshooting an intrusion detection dataset: the CICIDS2017 case study," in *2021 IEEE Security and Privacy Workshops (SPW)*, pp. 7–12, May 2021.
[12] A. Rosay, E. Cheval, F. Carlier, and P. Leroux, "Network intrusion detection: A comprehensive analysis of CIC-IDS2017," in *Proceedings of the 8th International Conference on Information Systems Security and Privacy*, SCITEPRESS - Science and Technology Publications, 2022.
[13] V. Kanimozhi and T. P. Jacob, "Artificial intelligence based network intrusion detection with Hyper-Parameter optimization tuning on the realistic cyber dataset CSE-CIC-IDS2018 using cloud computing," in *2019 International Conference on Communication and Signal Processing (ICCSP)*, pp. 0033–0036, Apr. 2019.
[14] M. V. O. Assis, L. F. Carvalho, J. Lloret, and M. L. Proença, "A GRU deep learning system against attacks in software defined networks," *Journal of Network and Computer Applications*, vol. 177, p. 102942, Mar. 2021.
[15] M. U. Ilyas and S. A. Alharbi, "Machine learning approaches to network intrusion detection for contemporary internet traffic," *Computing*, Jan. 2022.
[16] B. M. Serinelli, A. Collen, and N. A. Nijdam, "On the analysis of open source datasets: validating IDS implementation for well-known and zero day attack detection," *Procedia Comput. Sci.*, vol. 191, pp. 192–199, Jan. 2021.
[17] R. S. Arslan, "FastTrafficAnalyzer: An efficient method for intrusion detection systems to analyze network traffic," *Dicle Univ. Tip Fakul. Derg.*, vol. 12, no. 4, pp. 565–572, 2021.
[18] Q. R. S. Fitni and K. Ramli, "Implementation of ensemble learning and feature selection for performance improvements in Anomaly-Based intrusion detection systems," in *2020 IEEE International Conference on Industry 4.0, Artificial Intelligence, and Communications Technology (IAICT)*, pp. 118–124, July 2020.
[19] X. Li, W. Chen, Q. Zhang, and L. Wu, "Building auto-encoder intrusion detection system based on random forest feature selection," *Computers & Security*, vol. 95, p. 101851, 2020.
[20] J. Kim, J. Kim, H. Kim, M. Shim, and E. Choi, "Cnn-based network intrusion detection against denial-of-service attacks," *Electronics*, vol. 9, no. 6, p. 916, 2020.
[21] M. A. Ferrag, L. Maglaras, S. Moschoyiannis, and H. Janicke, "Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study," *Journal of Information Security and Applications*, vol. 50, p. 102419, 2020.
[22] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection," *Journal of Information Security and Applications*, vol. 54, p. 102564, Oct. 2020.
[23] M. Verkerken, L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: Unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, p. 12, Oct. 2021.
[24] A. H. Lashkari, G. Draper-Gil, M. S. I. Mamun, and A. A. Ghorbani, "Characterization of tor traffic using time based features," in *ICISSp*, pp. 253–262, 2017.
[25] "CICFlowMeter." https://github.com/ahlashkari/CICFlowMeter. Accessed: 2022-2-22.
[26] "IDS 2017." https://www.unb.ca/cic/datasets/ids-2017.html. Accessed: 2022-2-22.
[27] "IDS 2018." https://www.unb.ca/cic/datasets/ids-2018.html. Accessed: 2022-2-22.
[28] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, "A survey of network-based intrusion detection data sets," *Comput. Secur.*, vol. 86, pp. 147–167, Sept. 2019.
[29] A. Thakkar and R. Lohiya, "A review of the advancement in intrusion detection datasets," *Procedia Comput. Sci.*, vol. 167, pp. 636–645, Jan. 2020.
[30] J. L. Leevy and T. M. Khoshgoftaar, "A survey and analysis of intrusion detection models based on CSE-CIC-IDS2018 big data," 2020.
[31] J. V. V. Silva, N. R. de Oliveira, D. S. V. Medeiros, M. A. Lopez, and D. M. F. Mattos, "A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms," *Ann. Telecommun.*, Feb. 2022.
[32] L. D'hooge, T. Wauters, B. Volckaert, and F. De Turck, "Classification hardness for supervised learners on 20 years of intrusion detection data," *IEEE Access*, vol. 7, pp. 167455–167469, 2019.
[33] R. F. Bikmukhamedov and A. F. Nadeev, "Lightweight machine learning classifiers of IoT traffic flows," in *2019 Systems of Signal Synchronization, Generating and Processing in Telecommunications (SYNCHROINFO)*, pp. 1–5, July 2019.
[34] A. Sivanathan, H. H. Gharakheili, and V. Sivaraman, "Detecting behavioral change of IoT devices using Clustering-Based network traffic modeling," *IEEE Internet of Things Journal*, vol. 7, pp. 7295–7309, Aug. 2020.
[35] R. Doshi, N. Apthorpe, and N. Feamster, "Machine learning DDoS detection for consumer internet of things devices," in *2018 IEEE Security and Privacy Workshops (SPW)*, pp. 29–35, May 2018.
[36] M. Zolanvari, M. A. Teixeira, L. Gupta, K. M. Khan, and R. Jain, "Machine Learning-Based network vulnerability analysis of industrial internet of things," *IEEE Internet of Things Journal*, vol. 6, pp. 6822–6834, Aug. 2019.
[37] V. Morfino and S. Rampone, "Towards Near-Real-Time intrusion detection for IoT devices using supervised learning and apache spark," 2020.
[38] Huang, Qu, Jia, and Zhao, "O2u-net: A simple noisy label detection approach for deep neural networks," *Proc. IEEE*.
[39] L. Li, K. Jamieson, G. DeSalvo, A. Rostamizadeh, and A. Talwalkar, "Hyperband: a novel bandit-based approach to hyperparameter optimization," *J. Mach. Learn. Res.*, vol. 18, pp. 6765–6816, Jan. 2017.
[40] C. Northcutt, L. Jiang, and I. Chuang, "Confident learning: Estimating uncertainty in dataset labels," *J. Artif. Intell. Res.*, vol. 70, pp. 1373–1411, Apr. 2021.
[41] C. G. Northcutt, A. Athalye, and J. Mueller, "Pervasive label errors in test sets destabilize machine learning benchmarks," Mar. 2021.
[42] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *CoRR*, vol. abs/1611.03530, 2016.