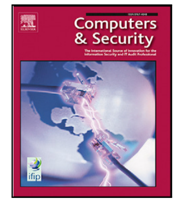
Computers & Security 156 (2025) 104510

Contents lists available at ScienceDirect

Computers & Security

journal homepage: www.elsevier.com/locate/cose



Full length article

Network intrusion datasets: A survey, limitations, and recommendations

Patrik Goldschmidt [a,b,*], Daniela Chudá [b,c]

[a] Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, Brno, 612 00, Czech Republic
[b] Kempelen Institute of Intelligent Technologies, Sky Park Offices, Bottova 7939/2A, Bratislava, 811 09, Slovakia
[c] Faculty of Electrical Engineering and Information Technology, Slovak University of Technology in Bratislava, Ilkovičova 3, Bratislava, 841 04, Slovakia

ARTICLE INFO

Dataset link: https://github.com/xGoldy/nid-datasets

Keywords:
Network intrusion detection
NIDS
Machine learning for intrusion detection
Cybersecurity datasets
NIDS best practices
Comparative dataset analysis

ABSTRACT

Data-driven cyberthreat detection has become a crucial defense technique in modern cybersecurity. Network defense, supported by Network Intrusion Detection Systems (NIDSs), has also increasingly adopted data-driven approaches, leading to greater reliance on data. Despite the importance of data, its scarcity has long been recognized as a major obstacle in NIDS research. In response, the community has published many new datasets recently. However, many of them remain largely unknown and unanalyzed, leaving researchers uncertain about their suitability for specific use cases.

In this paper, we aim to address this knowledge gap by performing a systematic literature review (SLR) of 89 public datasets for NIDS research. Each dataset is comparatively analyzed across 13 key properties, and its potential applications are outlined. Beyond the review, we also discuss domain-specific challenges and common data limitations to facilitate a critical view on data quality. To aid in data selection, we conduct a dataset popularity analysis in contemporary state-of-the-art NIDS research. Furthermore, the paper presents best practices for dataset selection, generation, and usage. By providing a comprehensive overview of the domain and its data, this work aims to guide future research toward improving data quality and the robustness of NIDS solutions.

1. Introduction

"*The only truly secure system is one that is powered off, cast in a block of concrete and sealed in a lead-lined room with armed guards – and even then I have my doubts.*" (Dewdney, 1989). The legendary quote by Eugene H. Spafford from 1989 is not any less relevant nowadays. Each year, cyberattacks like malware or denial of service (DoS) cause losses of trillions of USD (eSentire, 2023). With recent attacks on hospitals and medical devices, even people's lives are in danger (Papaioannou et al., 2022). Therefore, robust security is required to minimize an attack surface and mitigate potential harm.

In general, information technology security spans three tasks: prevention, detection, and reaction. As perfect prevention is unattainable and reaction implicitly assumes that the attack has already taken place and has been detected, a considerable focus must also be put on detecting cyber threats (Apruzzese et al., 2023b). This detection is done by intrusion detection systems.

An Intrusion Detection System (IDS) is a device or software application monitoring the activity of a target system or a computer network to identify their unauthorized use, misuse, or abuse (Mukherjee et al., 1994). Depending on the attack detection paradigm, we distinguish between signature-based (react on known attack patterns) and anomaly-based (react on unknown deviation from the norm) systems. Such systems can operate on the host level – Host-based IDS (HIDS) or the network – Network-based IDS (NIDS). Additional IDSs characteristics include the ability to operate in real-time and whether they merely report the attack or actively participate in its mitigation (Thakkar and Lohiya, 2022; Buczak and Guven, 2016).

An efficient IDS maximizes attack detection (ideally 100%) while minimizing the false alarm rate, i.e., legitimate samples flagged as malicious (ideally 0%). The performance of detection systems thus needs to be evaluated, analyzed, and compared using datasets containing malicious and legitimate samples. For this reason, public intrusion datasets play an important role in IDS research and development (Flood et al., 2024). The importance of data has been further emphasized by the rise of data-driven approaches for intrusion detection, most notably machine learning (ML) (Boutaba et al., 2018).

Despite the dire need for data, a general consensus within the network intrusion detection (NID) domain over the past decade was that there had been a lack of quality datasets for evaluating NIDS proposals (Abt and Baier, 2014; Małowidzki et al., 2015; Silva et al., 2022). This factor has significantly contributed to the poor adoption of ML-based methods into real-world systems despite huge scientific efforts (Sommer and Paxson, 2010).

* Corresponding author at: Faculty of Information Technology, Brno University of Technology, Božetěchova 1/2, Brno, 612 00, Czech Republic.
E-mail addresses: patrik.goldschmidt@kinit.sk (P. Goldschmidt), daniela.chuda@kinit.sk (D. Chudá).

https://doi.org/10.1016/j.cose.2025.104510
Received 11 February 2025; Received in revised form 6 April 2025; Accepted 26 April 2025
Available online 12 May 2025

As a response, the community has finally realized the importance of the data and put more effort into addressing data-related issues. However, the latest survey focused on listing, analyzing, and comparing network intrusion datasets was published in 2019 by Ring et al. (2019b). Since then, dozens of new datasets have been released, but no comprehensive study on NID datasets has been published. Many new datasets thus remain undiscovered and unanalyzed by the community.

As we show in this work, data scarcity may no longer be the primary issue. Instead, we argue that *datasets exist, but the community is largely unaware of their existence and lacks knowledge of best practices for their creation and usage.* This can be largely attributed to the lack of standardized NID data repositories and the neglect of domain-specific properties, leading to biased evaluations and over-optimistic results presented in many recent studies (Arp et al., 2022).

Contributions: Aiming to address the issues with discovering and handling data for network intrusion detection, we consider the primary contributions of the paper as follows:

1. We conduct a systematic, up-to-date comparative survey of public datasets for NIDS research and development published until 2023 (inclusive). This will introduce readers to the latest trends and possibilities for conducting experiments and method benchmarks. To the best of our knowledge, this survey covers the largest number of publicly available NID datasets (89).
2. In order to aid in data selection, we extract 13 properties for each dataset by manually downloading and analyzing the data. We also study the popularity of datasets in state-of-the-art NIDS research. To enhance accessibility, we share our analysis notebooks on GitHub,[1] allowing others to review dataset structures without having to download the full data.
3. We synthesize findings on flaws of existing datasets and describe their relation to domain-specific properties. In this manner, we elaborate on differences between NID and other ML application domains to promote a deeper understanding of data and, therefore, its more effective utilization.
4. Finally, we advocate the correct creation and usage of the data by discussing data handling recommendations.

The rest of the paper is composed as follows: We define the survey's scope and methodology in Section 2. Section 3 elaborates on the NID domain-specific properties and their impact on data. Section 4 describes 13 collected properties used to compare the datasets. The main part of this paper—the data survey—is presented in Section 5, listing datasets in a tabular format, categorizing them, and briefly elaborating on their properties and potential use cases. This section also analyzes dataset popularity and trending developments in NID data research. We discuss recommendations for data selection, creation, and usage in Section 6. Finally, Sections 7–9 outline future research directions, discuss related work, and conclude the paper.

2. Survey methodology

In order to ensure comprehensive, unbiased, and reproducible results, we followed the Systematic Literature Review (SLR) methodology for computer science (Kitchenham and Charters, 2007). This section outlines the survey's scope and research questions, with further details provided in Appendix A.

2.1. Scope of the survey

Our paper aims to provide a holistic view of the network intrusion datasets and their properties. Therefore, we do not narrow down on any specific environment or intrusion type but include datasets containing network traffic from various environments like the Internet of Things (IoT), Cyber-Physical Systems (CPSs), Software Defined Networks (SDNs), or Critical infrastructure Supervisory Control and Data Acquisition (SCADA).

Although not restricted to a particular environment, we strictly include only datasets containing network traces. Therefore, datasets without activity observable on a computer network (i.e., packets or flows) were excluded. As a result, the survey did not consider datasets that provide solely host-based data or environment-specific features like sensor values typical for CPS and SCADA datasets.

Given this scope, we emphasize that this survey is not supposed to be an exhaustive list of all available IDS datasets and, thus, should not serve as the sole data reference for specific use cases. Instead, we encourage consulting other surveys focused on specific cybersecurity subdomains as needed.

2.2. Research questions

Concerning the paper's goal and scope, we define our research questions as follows:

RQ1: What datasets are available for the network intrusion detection research and development?

RQ2: What are the key qualitative and quantitative properties of the identified datasets?

RQ3: Regarding the data properties from RQ2, are there any prevalent patterns or trends in dataset creation?

RQ4: What were the most popular datasets within the domain in the past years?

The given questions were addressed via a systematic review of scientific literature published until the year 2023 (inclusive) and results synthesis. After filtering, we downloaded, manually analyzed, and compared 89 relevant datasets for NIDS research. More information about this process, i.e., the search and selection of relevant studies, is described in Appendix A.

3. Domain-specific properties and their impact on data

The research on intrusion detection (ID) initially revolved around static signatures or simple statistical methods. Nevertheless, they require significant manual interventions, such as maintaining signature databases and threshold values. Motivated to minimize the need for manual labor and scale to big data more efficiently, the research community has gradually adopted data-driven approaches for ID. Nowadays, artificial intelligence, and especially machine learning, are crucial parts of ID research (Apruzzese et al., 2023a).

Despite its rapid adoption in research, ML penetrated into practical cybersecurity solutions much slower than in other domains, such as computer vision or natural language processing, which have been actively using ML for over a decade (Apruzzese et al., 2023a). In fact, only a fraction of real-world network monitoring and threat detection teams relied on machine learning in early 2020s (Alahmadi et al., 2022).

The discrepancy between academic research and the actual deployments of ML-based NIDSs was discussed by Sommer and Paxson (2010), who noticed that operators were reluctant to adopt ML-based anomaly detectors despite high research efforts within the area. This fact was attributed to fundamental differences between NID and other ML-application domains, driven by factors like very high cost of errors, enormous data variability, semantic gap (ML explainability issues), and difficulties with evaluation—amplified by the lack of benchmark data. Although published more than 10 years ago, most of these issues arose from the specific properties of cybersecurity and computer networking domains, affecting the research and development efforts to this day. This section elaborates on these problems and their impact on overall NID data availability and quality.

[1] https://github.com/xGoldy/nid-datasets.

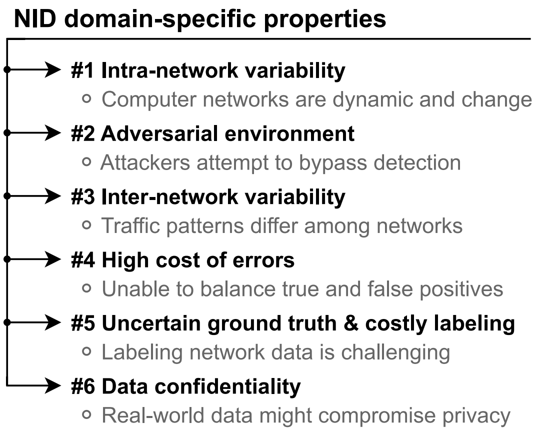


Fig. 1. Summary of Network Intrusion Detection domain-specific properties affecting data collection, handling, and interpretation.

3.1. When hell comes to earth: Unpleasant properties of the network intrusion detection domain

Network intrusion detection combines properties of both cyberthreat detection and computer networking domains. These properties significantly influence how the data are collected, handled, and interpreted. If machine learning is used for detection, an assumption on the data to be independent and identically distributed (*i. i. d.*) is also introduced (Dundar et al., 2007).

In essence, the *i. i. d.* assumes the data to be uncorrelated, while the data for training a model to be similar, i.e., drawn from the same distribution, as the future observed data. If not met, the model's performance might be negatively affected. However, the intrinsic properties of cyberthreat detection and computer networking interfere with this assumption and thus impair ML deployment in real-world scenarios (Apruzzese et al., 2023a). In the following paragraphs, we dive deeper into these properties and outline their potential impacts, as summarized in Fig. 1.

(1) *Network traffic diversity and dynamicity (intra-network variability)*: Computer networks are highly dynamic by nature. Network environments constantly change as new devices are connected to the network, new applications and protocols are developed, or even new networking paradigms, such as the Internet of Things (IoT), are introduced. The temporal aspect further introduces periodicity patterns based on day-and-night cycles and office hours (Haffey et al., 2018).

(2) *Adversarial environment*: In addition to natural networks' dynamicity, the NID domain has to assume the implicit presence of adversaries. Due to the endless cybersecurity arms race, attacks continuously evolve, and new vulnerabilities are discovered. Moreover, the attackers may also attempt to bypass detection via adversarial samples crafted with obfuscation techniques or tiny perturbations of the input data (He et al., 2023).

(3) *Inter-network variability*: Every network is unique (Sommer and Paxson, 2010). Each has its specific users, devices, topology, and Internet service provider. For these reasons, traffic patterns may vary significantly among distinct networks. An attack with the same parameters executed in different networks will likely vary in its effects and observed traffic characteristics.

(4) *High cost of errors*: Unlike other ML applications, the relative cost of any output error (i.e., misclassification) is very high. Regular applications can optimize for lower false alarm (false positive) or higher detection (true positive) rates. However, NID needs to consider both. Even a single missed anomaly might lead to a successful intrusion with dire consequences. On the other hand, a low relative false alarm rate can still create a tremendous amount of actual alarms due to a traffic imbalance, known as base-rate fallacy (Axelsson, 1999). For instance, with a 99% true positive and 1% false positive rate in a 1:100 class balance, we receive 100 false alarms for every 99 true attacks.

(5) *Uncertain ground truth and costly labeling*: In many domains, e.g., computer vision, the ground truth is clear and stable: "A cat will always be a cat, whereas a dog will always be a dog" (Apruzzese et al., 2022), and non-experts can often distinguish between classes very well (Law and von Ahn, 2011). Moreover, data augmentation (Shorten and Khoshgoftaar, 2019) can increase the effectiveness of existing labeled data. However, determining the ground truth in NID is complex. Due to its adversarial nature, a sample benign today might be malicious tomorrow (label shift). High inter-network variability can cause anomalies in one network to be considered benign elsewhere.

In NID, even domain experts might struggle to verify ground truth, whereas attack detectors might return different and even contradicting results (Charlton et al., 2018). These facts disable crowdsourcing annotation, a common way to obtain large corpora of labeled data (Zhang et al., 2016). Finally, even all the traffic of infected hosts cannot be considered malicious, as some might correspond to legitimate activities (e.g., ARP messages) (Apruzzese et al., 2022). For these reasons, accurately labeling network data is difficult and often infeasible.

(6) *Data confidentiality*: Captures from real networks might contain sensitive and confidential information, such as IP addresses or packet payloads, significantly limiting their sharing. Although anonymization techniques (Coull et al., 2009), like IP address remapping or payload stripping, can be used, they decrease the data's value—e.g., disabling payload-analysis NIDSs.

Given the challenges of labeling and data sharing, the scarcity of high-quality real-world data becomes understandable. However, the problem persists even when such data are collected and labeled. Since data is captured at a specific point in time, future changes in the underlying distribution (data drift) or a relationship between input features and the output labels (concept drift) violate the *i. i. d.* assumption. This phenomenon, known as model drift, leads to a gradual decline in a prediction model's performance over time. Since NID is adversarial and highly dynamic, model drifting, amplified by the low tolerance of errors, is considered a major obstacle in ML utilization in real-world scenarios. While approaches like incremental (Masana et al., 2023), lifelong (Parisi et al., 2019), or active learning (Settles, 2009) have been proposed to mitigate drift by leveraging newly observed data for model updates, applying ML for real-world NIDSs remains heavily constrained.

3.2. Limitations of existing datasets

Despite the mentioned obstacles, various datasets for evaluating NIDSs have been proposed over the years. Nevertheless, all of them were directly or indirectly influenced by the domain properties, resulting in several limitations. Since these properties are unlikely to change, future datasets will likely exhibit some of these flaws as well, and no dataset will ever be "perfect" (Ring et al., 2019b). In the following paragraphs, we elaborate on these limitations (summarized in Fig. 2).

(1) *Timeliness*: The direct consequence of constantly evolving network and threat landscapes is data timeliness. Generally, it cannot be assumed that the behavior collected during a certain period will remain immutable over time (Viegas et al., 2017). Therefore, static datasets become less representative of real-world scenarios as they age (Catillo et al., 2023), affecting the soundness of the evaluation results. Although continual updates with the newest traffic patterns would mitigate this issue, immutable (static) datasets are required for fair benchmark comparisons between systems.

(2) *Real traffic limitations*: If the dataset authors want to utilize real-world traffic, data anonymization is required to ensure privacy. Nevertheless, anonymization decreases data realism and disables various use cases (e.g., payload analysis) (Abt and Baier, 2014). Furthermore, capturing data from real-world enterprise networks might be unfeasible due to excessive traffic volume. For this reason, traffic sampling might need to be introduced, causing additional bias or incomplete network flows (Silva et al., 2017; Meng et al., 2018).

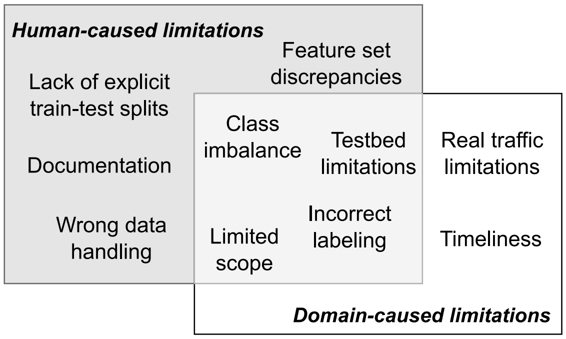


Fig. 2. Limitations of datasets for network intrusion detection. As illustrated, some are a direct consequence of domain-specific properties and are challenging to address (e.g., timeliness), some are primarily caused by a human factor and can be mitigated completely (e.g., documentation), and those that lay in an intersection of both can be addressed only partially (e.g., class imbalance, as NID traffic is naturally imbalanced, but data authors might attempt to reduce it).

(3) *Testbed limitations*: Due to the limitations of real-world data, most public datasets contain emulated traffic captured in a lab testbed environment. Despite some benefits of such an approach, its data are not guaranteed to resemble realistic network behavior and threats (Molina-Coronado et al., 2020). Such datasets tend to model over-simplified network conditions, failing to capture their real-world complexity and diversity (Catillo et al., 2023; Flood et al., 2024). The lack of realism in benchmark data is considered a major obstacle in deploying anomaly-based NIDSs in real-world environments (Sommer and Paxson, 2010; Viegas et al., 2017). For instance, Catillo et al. (2021b) have shown that many DoS attacks from public datasets do not resemble real-world characteristics due to being inefficient (not achieving their malicious intent) or assuming non-realistic environment configurations.

(4) *Class imbalance*: Modern machine learning classifiers are trained to maximize predictive accuracy by minimizing misclassification errors (Thabtah et al., 2020). Such a setting usually assumes roughly balanced classes, i.e., with similar prior probabilities. However, cyberthreat detection domains like NID typically deal with a huge imbalance in favor of the negative (benign) class, as only a few attack instances are observed along a vast amount of non-malicious communication (Catillo et al., 2023). Although testing data should resemble real-world distribution to prevent spatial bias (Pendlebury et al., 2019), training a supervised classifier should be done on approximately balanced data. Nevertheless, most datasets are imbalanced, leaving to address the issue by their users (Bagui and Li, 2021a).

(5) *Documentation*: The dataset's accompanying metadata and documentation are crucial in determining the data value and its suitability for particular use cases. However, datasets often lack the documentation completely or omit necessary details. For this reason, the soundness of conclusions drawn by the data users is significantly hampered.

(6) *Lack of explicit train–test splits*: Many datasets are not explicitly split for training and testing. As a result, their users create arbitrary splits, causing their methods to be incomparable, as each may be trained and evaluated on a different data subset. Furthermore, applying methods like cross-validation or partitioning the data without respecting time dependencies may lead to temporal bias and inflated evaluation performance (Arp et al., 2022; Pendlebury et al., 2019).

(7) *Feature set discrepancies*: As shown in our survey, many dataset authors propose their own feature sets, typically based on network flows. While some provide only 10 – 15 essential features, others offer over 100 advanced statistics. Several newer datasets adopt the CICFlowMeter tool (Lashkari, 2016) for feature extraction. An attempt to unify feature sets was made by Sarhan et al. (2022b, 2021), who proposed extracting 43 NetFlow v9 features using nProbe (Ntop, 2023) flow collector. Despite these efforts, the community has not yet agreed on a unified, widely accepted feature set, hampering the transferability and generalizability between datasets with different features.

(8) *Reproducibility:* In computer networking, it is infeasible to collect exactly the same data twice, even in identical environments. This property disallows the reproducibility of the raw packet data. Although it would not be a critical limitation by itself, reproducibility is further restricted for datasets publishing only extracted features (e.g., network flows) without packet data. Even if some datasets do provide raw data, they rarely publish source code and configuration for feature extraction and labeling. In all cases, reproducing the dataset (or even extracting the same features from packets) is infeasible, disallowing data validation and its possible extensions (Landauer et al., 2023).

(9) *Limited scope*: Due to the diversity of network technologies and cyberthreats, assembling a comprehensive dataset that covers the entire NID landscape is impractical. While specialized datasets (e.g., IoT) have been released recently, many scenarios and intrusions remain underrepresented. In particular, Hindy et al. (2020) note that public datasets cover only 33% of known attacks. Given that new attacks and vulnerabilities continually emerge, NIDSs benchmarked on static datasets will never fully comprehend all potential threats in the wild.

(10) *Incorrect labeling*: Due to networking dynamicity and the common methods for traffic labeling (based on the attack time and the source and destination addresses of the attacker/victim (Guerra et al., 2022)), NID data are prone to mislabeling. The issue has been demonstrated by Liu et al. (2022) and Flood et al. (2024) through analyses of several popular datasets. Such mislabeling can result in unrealistic results and benchmark destabilization (Northcutt et al., 2021).

(11) *Methodically wrong data handling*: In addition to mislabeling, several other factors, primarily originating from human error in data collection and processing, affect the final dataset. A common issue we encountered was improper timestamping, as some datasets contain incorrect timestamps or time inconsistencies of events (e.g., attacks) with the documentation. We also observed disordered packets, undocumented capture gaps, and capture interruptions, resulting in improperly terminated files. Data processing issues, often due to bugs in feature extractor software (Catillo et al., 2023), include missing values, duplicated features or entries, and zero-variance features (e.g., all zeros). These issues are frequently unreported, creating ambiguity for users and undermining the soundness of their conclusions.

While many discussed limitations arising from the domain-specific properties are challenging to address, others (documentation, lack of explicit splits, feature set discrepancies, reproducibility, and data handling errors) are entirely within the control of dataset authors. Even when a limitation cannot be fully mitigated, proper documentation reduces ambiguity and significantly enhances data usability. For this reason, we consider documentation a crucial pillar of every dataset. However, most do not discuss any limitations at all. Supposing that authors do not release inherently flawed records on purpose, we theorize that they often neglect thorough analysis of their collected data.

From the perspective of data users, understanding these limitations aids in efficient data selection, proper handling, and avoiding evaluation errors. In Section 6, we discuss recommendations to minimize or entirely prevent their impact.

4. Network intrusion data and its properties

Depending on the dataset goal and capture attributes, we characterize network intrusion data by various properties that influence their use cases and the soundness of methods utilizing them. In this survey, we collected 13 essential properties to provide an unbiased, comprehensive view and comparison of available datasets for NID purposes. A portion of them was adopted from Ring et al. (2019b)'s survey.

We split the extracted information into five categories: General information, Nature of data, Data volume, Network properties, and Evaluation. This section elaborates on these categories and individual properties to aid in data understanding, selection, as well as facilitate the evaluation of datasets not included in this survey in future research. Their summary is provided in Table 1.

Table 1
Summary of 13 collected properties of the surveyed NID datasets split into five categories. The table briefly describes each property, and the fourth column ("Possible values") is a comma-separated list of values a particular property can attain. Curly brackets symbolize a set from which a single value can be drawn, whereas × is a Cartesian product of such sets in cases when multiple elements define a single property. Ampersand ("&") symbol is occasionally used to combine multiple values within a single entry.

Category	Property	Description	Possible values
General information	Year	Year of the data collection	{1998, 1999, …, 2023}, Now[a]
	Focus	Focus (scenario) of the dataset	text
	Normal	Normal (benign) traffic type	Real , emul-{p, v, pv}, synth, no
	Attack	Attacking traffic type, number of attacks, and attack categories	{real, emul-{p, v, pv}, synth} × (ℕ) × Attack types (see Table 2)
Nature of data	Format	Format of the data	packet ({bin, txt}), flows ({uni, bi}), logs ({net, host, both}), nethost, other
	# Features	Number of dataset features	$\{n \mid n \in \mathbb{N}\}$, "–" for raw data formats
	Anonymized	Anonymized parts of the dataset	No, IPs, MACs, payload, time, names, n. s.[b]
Data volume	Size	The amount of records and/or packets. Train and test sets split by the '+ ' sign.	$\mathbb{R}^+$ {k, M, …}[c] × {pkts, flows, recs.}
	Duration	Data capture time length and its continuousness. Train/test split by the '+' sign. Periodicity expressed with 'x' (times).	$\{n \mid n \in \mathbb{R}^+\}$ × {m, h, d, w, mo}[d] ×{cont., discont., periodic}
Network properties	Network type	Type of the network environment	{small, medium, large} ×{academic, enterprise, industrial, military, wireless, cloud, honeypot, IoT, SDN, P2P, ISP, ∅}, mixed, n. s.
	Complete capture	Traffic from the entire network is provided	Yes, No, n. s
Evaluation	Split	Train and test subsets are pre-split	Yes, No
	Labels	The ground-truth is provided	Yes, No, Indir.

[a] Signifies the capture was still ongoing at the time of writing.
[b] "n. s." is an abbreviation for "not specified".
[c] Metric suffixes, i.e., k – 10^6, M – 10^9.
[d] Time units (m – minute, h – hour, d – day w – week, mo – month).

4.1. General information

The primary considerations for dataset selection include analysis of its scenarios along with other details like the collection period, the types of attacks, and the environment used to perform the capture:

Year: The specific year(s) or period of data collection. Since many datasets are published months or years after collection, reporting the actual collection date is crucial for understanding the captured threat landscape and traffic characteristics (e.g., popular applications). For continuously ongoing captures, such as MAWILab (Fontugne et al., 2010), we denote the value as *Now*.

Focus: This property summarizes the dataset scenario and its potential use cases. Based on the focus, we distinguish between general- or special-purpose datasets. General-purpose datasets (*General*) are captured in traditional local area or enterprise networks with various intrusive scenarios. Depending on the nature of the data, we include *IDS* (both host and network information) or *NIDS* (only network data are present) identifiers. In contrast, special-purpose datasets focus on a particular attack like DDoS or a specific environment such as IoT.

Normal: Indicates whether normal (benign) traffic is present and how it was generated. If absent, the value is recorded as *no*. Otherwise, we distinguish between data obtained from *real*, emulated (*emul*), or synthetic (*synth*) environments. Since their various definitions can be found in the literature (especially, emulated and synthetic terms are often used interchangeably), we establish the following definitions:

- *real* – Traffic originating from real-world operational networks captured using tools like `tcpdump` or `nfdump`. It reflects genuine network interactions with purposes beyond traffic generation. Since it contains real information, it must be anonymized. Additionally, real data cannot be declared benign with absolute confidence, as undetected attacks might be present. Instead, it is referred to as *background* traffic without evident anomalous or malicious behavior.
- *emul* – Emulated traffic is captured in a network environment the same way as *real* traffic, however, with a different purpose. While *real* traffic encompasses genuine network interactions (e.g., web browsing), the primary purpose of emulation is to simulate and capture a specific behavior via activity simulation scripts, automation frameworks like Selenium,[2] or traffic generators. We further distinguish between three types of emulated traffic depending on the capture environment:
  - *emul-p*: Traffic from physical networks with real network equipment like routers and end-point stations.
  - *emul-v*: Traffic from virtualized networks and virtual machines without their actual physical representation, e.g., NS3 (Riley and Henderson, 2010), Mininet (Lantz et al., 2010), or cloud-based networks.
  - *emul-pv*: Traffic from mixed environment combining physical equipment with virtualized elements.
- *synth* – Similar to emulation, synthetic traffic is also designed for the sole purpose of simulating specific behaviors. Nevertheless, whereas emulation creates and captures actual packets traversing a network, synthesis implies no network activity. Instead, the data is generated by models that mimic activities without their actual network representation.

We highlight these three traffic origins due to differences in data characteristics, especially realism. In general, real-world traffic is considered ideal as it offers genuine network interactions. However, it suffers from mentioned privacy concerns and may lack coverage of specific events, such as flash crowds (Ari et al., 2003), if they are not present during the capture.

In contrast, emulated and synthetic traffic offer greater flexibility, lower collection costs, and more reliable labeling. However, as outlined in Section 3.2, their crucial limitation is realism. Due to the complexity of network traffic, replicating real-world scenarios in a laboratory environment is very challenging (Catillo et al., 2023). Although several works have tried to tackle this issue, e.g., β-profile system (Shiravi et al., 2012), realistic simulation of network behavior remains an important research direction (Hindy et al., 2020).

While traffic generation methods significantly impact realism, the environment type also matters. In general, physical deployments for traffic simulation are considered more realistic than virtual ones (Chou and Jiang, 2020), although the differences might be minor.

[2] Selenium is a framework for browser automation and web page functional testing. Homepage: https://www.selenium.dev/.

Table 2
The scheme for a quick reference of attacks present in the reviewed datasets. Each attack category is assigned an acronym, such as "B" for Brute-force. A sequence of such acronyms (e.g., BCO) then represents the presence of given types of attacks within the dataset.

Attack category	Acronym	Description
Brute-force	B	Brute-force performs trial-and-error attempts to guess login credentials, passwords, or encryption keys. These trials are typically based on pre-compiled lists and dictionaries.
C&C Botnet traffic	C	Command and Control (C&C) traffic is used by infected machines (bots) to communicate and receive orders from the master server within the client–server botnet architectures.
Denial of service	D	Denial of Service (DoS) and Distributed DoS (DDoS) are deliberate attempts to disrupt a target machine or network, causing unavailability for regular users.
Reconnaissance	R	Reconnaissance (scanning, probing) is a process of gathering information about potential targets, vulnerabilities, and attack vectors.
Vulnerability exploitation	V	The act of exploiting various vulnerabilities, such as SQL injection or buffer overflow, typically to gain remote access or escalate privileges on the target machine.
Other	O	Environment-specific or uncommon attacks present in a minority of datasets—e.g., SPAM, data exfiltration, ransomware, cryptomining, or ARP spoofing.
Unspecified	?	Signifies that some of the previous attack categories may be present but are not explicitly acknowledged. This happens when the dataset contains data from a real network, so the presence of intrusion attempts in the data is uncertain.

Naturally, synthetic traffic created without any network activity is considered the least realistic. However, its flexibility and lack of privacy concerns make it a promising research direction to tackle highly dynamic and ever-changing domains like intrusion detection. Synthetic traffic has already shown promising results in generating normal traffic with statistically similar properties to more realistic data (Schoen et al., 2024). Section 5.5.2 discusses trends in traffic generation in detail.

Attack: This property specifies three aspects regarding intrusions in the dataset: (1) Origin environment, (2) The number of distinguishable attacks, and (3) Attack types.

The attack origin environment is defined the same as for normal traffic: *real*, *emul-{p, v, pv}*, and *synth*. Real attacks correspond to traffic with actual malicious intent captured in real-world production networks, i.e., by detecting and sampling an ongoing attack, as in the Booters dataset (Santanna et al., 2015). Emulated attacks are launched in controlled environments just to simulate attacks' behavior, usually using the same tools as real cyber-attackers (e.g., in Kali Linux[3]). Finally, synthetic attacks were not captured in a network but rather generated by specialized algorithms.

Although in between real and emulated categories, we classify traces of sandboxed malware connected to the Internet (e.g., CTU-13 (García et al., 2014)) as real traffic, assuming its source code was not altered. In this case, we consider its patterns (e.g., spreading, Command & Control traffic) representative of real malware even when executed in a controlled environment.

The environment origin is followed by a value in brackets indicating the number of distinguishable attacks. This represents the number of malicious classes clearly separable in the data (either on a feature-based level based on labels or via raw packet filtering). For datasets like CUPID (Lawrence et al., 2022) that declare multiple attacks but provide only binary labels, i.e., distinguishable anomalies, the value is listed as one. Datasets with unspecified attack counts, e.g., MAWILab (Fontugne et al., 2010), have no associated value.

In the third part, we summarize the attack types to clarify the dataset's objectives and associated traffic. Although existing cyberattack taxonomies (Santos et al., 2025; Derbyshire et al., 2018) could be used, we developed a custom scheme (Table 2) to provide a concise summary of common attack events in NID datasets. Therefore, rather than a comprehensive taxonomy, it serves as a quick reference for identifying attacks within datasets. In contrast to other taxonomies that aim for completeness and specificity (Santos et al., 2025), our scheme intentionally includes the *Unknown* category to account for potentially undiscovered attacks, as well as the *Other* category for less common ones. Each attack type is then represented by an acronym (e.g., "D" for a (D)DoS attack), with multiple types forming a single acronym string.

For instance, the string *emul-v (10) DRO* for the Bot-IoT dataset (Koroniotis et al., 2019) indicates 10 attack classes generated by emulation in a virtualized environment. They belong to (D)DoS, Reconnaissance, and Other categories due to data theft and keylogging traces.

4.2. Nature of data

Three properties in this category describe how the data are stored and whether they have been preprocessed.

Format: Network monitoring is primarily based on collecting three types of data: packets, network flows, and logs (Zhou et al., 2018). Accordingly, NID datasets are typically distributed in these formats. Some also offer auxiliary data like end-host logs, routing tables, or SNMP databases. This survey outlines the four most common data formats and includes the category of *other* for specific cases, as described below:

- *packet*: Data provided in a per-packet manner. Most such datasets contain raw (binary) packets distributed as Packet Capture (PCAP) files. However, some (e.g., AWID2 Kolias et al., 2016) provide already pre-extracted full packets or their specific fields in a textual format. We distinguish them by the *packet (bin)* and *packet (txt)* identifiers.
- *flows*: A flow is a sequence of packets with some common properties that pass through a network device (Claise, 2004). These properties are typically considered as a 5-tuple: source IP, destination IP, source port, destination port, and a transport protocol identifier, used for packet aggregation. Aggregation can either be uni-directional (*flows (uni)*) or bi-directional (*flows (bi)*), recording both sides of the connection in a single flow entry.
  Collected traffic features depend on flow exporters (Vormayr et al., 2020), including general-purpose tools like YAF (Inacio and Trammell, 2010), nProbe (Ntop, 2023), and NFStream (Aouini and Pekar, 2022), as well as security-focused tools like Argus (Qosient, 2022), Zeek (Paxson, 1999), and CICFlowMeter (Lashkari, 2016). These exporters rely on various flow protocols, primarily NetFlow v5/v9 (Claise, 2004), IPFIX (Aitken et al., 2013), sFlow (Phaal and Lavine, 2004), and OpenFlow (Open Networking Foundation, 2015). While essential flow features (Table 3) are commonly found in most flow-based datasets, they are often insufficient for cybersecurity applications. As a result, extended versions with additional statistics, such as NetFlow v9 (NetFlow Version 9, 2011) or CICFlowMeter (Lashkari, 2021) feature sets, are commonly used. Most frequently, these datasets are distributed in a textual comma-separated-values (CSV) format.

[3] Kali Linux is a Linux distribution designed for digital forensic analysis and penetration testing. Homepage: https://www.kali.org/.

Table 3
Common features for uni-directional flow datasets. Note that bi-directional datasets include separate fields for features 8–11, tracking source → destination and destination → source traffic separately, resulting in 15 features. Additional features may be added for data labels. Identifiers and some features might vary across datasets and protocols—e.g., NetFlow v9 (Claise, 2004) uses `FIRST_SWITCHED` as the first flow timestamp and `LAST_SWITCHED` for the last instead of defining duration explicitly. Nevertheless, despite naming differences, feature semantics remain identical across datasets.

#	Identifier	Description
1	`stime`	Timestamp of the first flow packet
2	`dur`	Flow duration
3	`srcip`	Source IP address
4	`dstip`	Destination IP address
5	`srcport`	Source transport protocol port
6	`dstport`	Destination transport protocol port
7	`proto`	Transport protocol
8	`tos`	Type of service
9	`pkts`	Destination to source packet count
10	`bytes`	Destination to source bytes sum
11	`flags`	Cumulative OR of TCP flags

- *logs*: In addition to data directly captured from a network (packets or flows), many practical IDSs utilize logs and alerts from other systems to improve detection capabilities or minimize false alarms. Therefore, some datasets also provide event logs from end-host systems (*logs (host)*), network (*logs (net)*), such as firewall logs or signature-based NIDS alerts, or from both sources at once (*logs (both)*). They can be used for indirect data labeling or evaluation of systems working with multiple data sources.
- *nethost*: Instead of providing separate captures from a network and logs from end-host devices, some datasets provide pre-extracted feature sets combining the two. For instance, a dataset feature set can consist of flow data enriched with statistics from the host system of interest, such as a server that the attackers target. These datasets enable faster experimentation by eliminating the need for a custom feature extractor from multiple data sources, but at the cost of decreased flexibility.
- *other*: Specific data formats not corresponding to any of the above. An example of this case is UNR-IDD (Das et al., 2023), which provides feature vectors as per-port statistics.

In general, packet-based datasets provide more detailed information because they provide exact contents and timestamps of every packet in contrast to flow aggregates. If packet payloads are not stripped, they also enable deep packet inspection (DPI) (Finsterbusch et al., 2014), a technique for full packet analysis. However, with increasing traffic volumes and widespread encryption, flow-based NIDSs have gained popularity for being unaffected by encryption and offering higher traffic throughput (Umer et al., 2017; Sperotto et al., 2010).

The number of features (# features): Defines the length of the data feature vector, including labels. This information helps to determine whether the dataset uses a basic feature set as in Table 3 (i.e., 10–18 NetFlow features) or an extended one. Note the value is only relevant for datasets with pre-extracted features.

Anonymized: In some cases, such as captures from real-world networks, the data require anonymization to preserve privacy or hide network details. This property lists such anonymized parts of the data. While anonymizing IP addresses via prefix-preserving algorithms like CryptoPan (Xu et al., 2002) is typically sufficient for flow-based data, full packet captures require additional actions. Oftentimes, MAC addresses are also obscured, and packet payloads trimmed to a certain length (e.g., to retain application headers) or removed entirely.

4.3. Data volume

This category evaluates the volume of the data from two perspectives: the number of data points (i.e., raw packets or feature vectors) and the data capture duration.

Although this information is typically declared in dataset documentation, we did not rely on it. Instead, we downloaded and analyzed the data ourselves due to occasional inconsistencies between the data and its documentation. For datasets with predefined training and testing splits, we report these volumes separately using a "+" sign. The properties are defined as follows:

Size: Specifies the number of packets (*pkts*), flows (*flows*), or records (*recs*) in the dataset, using metric suffixes kilo (k) = 10^6 and mega (M) = 10^9. A comma sign (",") represents a delimiter between thousands. In order to keep the table concise, we omit listing the size of associated log files, as the survey is focused on network data.

Duration: This property represents the dataset capture duration. While dataset documentation often defines duration as the time span between the first and last data timestamps, this approach may be misleading due to the presence of potential temporal gaps. Therefore, this survey aims to report the actual duration by disregarding such gaps.

Supposing the dataset entries are sorted chronologically, with t_i as a timestamp of record x_i and a difference between two adjacent records as $\tau_i = t_i - t_{i-1}$, a temporal gap in the data occurs if $\tau_i > \epsilon$. If no temporal gap exists ($\forall i : \tau_i \leq \epsilon$), i.e., the network activity is present throughout the entire dataset time span, we refer to the dataset as continuous (*cont.*). Otherwise, a dataset is discontinuous (*discont.*) with at least one capture gap ($\exists i : \tau_i > \epsilon$). In this matter, an actual capture duration D (Eq. (1)) is calculated by ignoring gaps longer than ϵ.

$$D = \sum_{i:\tau_i \leq \epsilon} \tau_i. \tag{1}$$

In this survey, data continuity is based on ϵ relative to the dataset span (the first and last timestamps difference), as, for instance, a 10 min gap is more acceptable in a month-long dataset than in a one-hour dataset. Therefore, we set ϵ as 1% of the analyzed dataset's absolute time span. Additionally, assessing continuity for datasets with a predefined train–test split is performed separately with different ϵ for each split, as merging them would be semantically incorrect.

A special case of continuity is periodicity (*periodic*), which refers to continuous blocks of network activity recurring in a regular pattern, such as specific hours per day or days per week. Periodic captures are noted as *periods x duration*. For instance, the DARPA 1998 dataset (Lippmann et al., 2000a) was captured over nine weeks on weekdays. Since it also contains a train–test split, it is listed as *7x5d + 2x5d periodic*, indicating seven five-day periods for training and two for testing, with each period being continuous.

In addition to determining how the data duration is computed, information about continuity helps assess the data suitability for detection systems relying on time dependencies between samples. If the data was discontinuous, these systems cannot leverage their dependency-analysis mechanisms, likely resulting in biased evaluation results.

For the practical analysis, we utilized `capinfos` and `tshark` tools for raw PCAP data and `pandas` for CSV files. We relied on duration from data documentation if the dataset did not offer a raw PCAP and its extracted features did not contain timestamps. In cases where no duration could be inferred from data and its documentation, the value is listed as "n. s." (not specified).

4.4. Network properties

The General information category distinguished the character of environment as real-world, emulated, or synthetic for both types of traffic. In this category, we provide additional information about the environment—the underlying network type and whether the capture covers the entire network.

Network type: This property defines the network's size and type to indicate expected traffic patterns. We categorize network sizes into three groups: (1) *small* (up to 99 hosts) – a small office/home office (SOHO), small company, or a household network, (2) *medium* (100–499 hosts), and (3) *large* (500+ hosts) – a large enterprise or an internet service provider (ISP). In addition to size, we also report the network's type, with the most common being *academic*, *enterprise*, *industrial*, *military*, *wireless*, *cloud*, *honeypot*, internet of things (*IoT*), software-defined network (*SDN*), peer-to-peer (*P2P*), and internet service provider backbone (*ISP*). This information is omitted if no specific type is provided in the documentation. Datasets with traces combined from multiple networks are assigned the value of "*mixed*".

Complete capture: Specifies whether the dataset contains data from the entire underlying network. In general, it is desirable to capture all network communication to ensure traffic diversity and realism. However, some datasets capture only from a single host (e.g., honeypot) or a specific set of hosts due to privacy or practical reasons. Such a setup might skew the traffic patterns (e.g., honeypots would receive significantly more malicious traffic than regular hosts (Sethuraman et al., 2023)). Therefore, specialized evaluations or NIDS types should be employed when dealing with incomplete captures.

4.5. Evaluation

Finally, this category groups properties influencing NIDS evaluation, such as whether the data have already been pre-split for training and testing and whether the dataset is labeled.

Split: Defines whether the dataset offers pre-split train and test subsets. This property is desirable, as it allows transparent comparisons of different NIDSs evaluated on the same data. On the other hand, datasets without a train–test split have to be split by their users, potentially leading to evaluation bias and selective reporting (Arp et al., 2022; Lipton and Steinhardt, 2019).

Labels: Specifies whether the dataset is labeled—a crucial property for NIDS comparison and analysis. We assign the value *yes* only if an unambiguous mapping between data instances and classes exists. This includes direct labels in feature vectors, as well as class separation via filters (e.g., IP-based packet labeling in 2017-SUEE8 Lukaseder, 2017) or directory structures (e.g., CAIDA DDoS 2007 CAIDA, 2010).

In addition to being labeled or unlabeled (*yes/no*), datasets can also be labeled indirectly (*indir.*). In this case, no direct mapping between records and classes exists, but the traffic is differentiated alternatively, such as with IDS alerts or log files. While this provides some level of ground truth, it is hardly completely precise, requires additional effort from users to correlate logs with the data, and can lead to ambiguous interpretations. As a result, evaluating with indirectly labeled or unlabeled data is error-prone and should be approached cautiously.

This section has presented 13 properties of NID datasets across five categories. We used these properties to establish an extraction protocol and collect data from the reviewed datasets. The survey results are presented in the following section.

5. Network intrusion datasets

This section, the core of this survey, lists and comparatively analyzes datasets for network intrusion detection. As outlined, the datasets were selected following the systematic literature review methodology discussed in Section 2 and Appendix A.1. Afterward, 13 properties defined in Section 4 were extracted through documentation review and data exploratory analysis. The analysis results—Jupyter Notebooks and scripts—are available at our GitHub repository (see Section 1).

The survey results are summarized in Table 4, with each row representing a single dataset. In rare cases like NF-UQ-NIDS (Sarhan et al., 2022b), two related datasets are listed as one entry divided by a "/" symbol. The entries are sorted in ascending order by year and then alphabetically. Datasets spanning multiple years are ordered based on their last capture year but appear before datasets captured in a single year. Dataset download links (functional in early 2025) are available in their associated references.

The following subsections analyze the surveyed datasets in more detail. In order to group the data, we developed a taxonomy (Fig. 3) classifying datasets as general-purpose (Section 5.1) or special-purpose

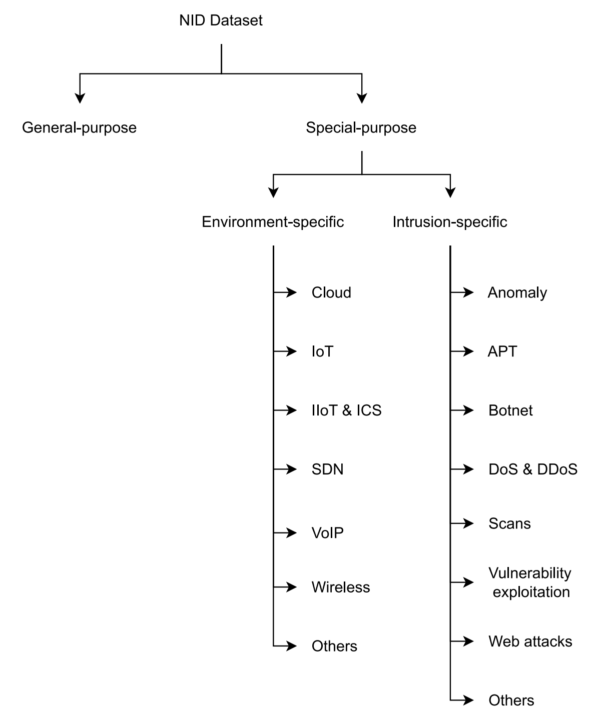


Fig. 3. Taxonomy of datasets for Network Intrusion Detection based on their focus. Note that special-purpose categories are not exclusive, as a single dataset can both focus on a particular intrusion type captured in a specific environment.

(Section 5.2) based on their focus and intended use cases. Afterward, Section 5.3 outlines sources for other datasets not included in this survey. In Section 5.4, we analyze the popularity of datasets in contemporary state-of-the-art NIDS research. Finally, Section 5.5 highlights key trends observed in network intrusion data research.

5.1. General-purpose datasets

General-purpose datasets, labeled as *General*, provide data for benchmarking intrusion detection systems for traditional environments such as local area networks (LANs) or enterprise networks. They offer a diverse set of intrusions rather than focusing on specific attacks.

Although general-purpose datasets cannot be precisely divided into subcategories without ambiguity (Fig. 3), we organize this subsection into five parts. Initially, we describe the first-generation datasets produced by DARPA. Next, we examine CIC datasets, which have largely replaced DARPA in recent research. Finally, we review other general-purpose datasets, including NetFlow, newly released, as well as already established but less commonly utilized.

5.1.1. DARPA & KDD

Datasets published by the Defense Advanced Research Projects Agency (DARPA) are considered the first generation of public IDS benchmarks. The first dataset, *DARPA 1998* (Lippmann et al., 2000a), emulating an Air Force base with thousands of hosts, includes a robust benign traffic profile along with 32 different attacks, categorized as Denial of Service (DoS), Probing, User-to-Root (U2R), and Remote-to-Local (R2L). Despite having different goals (escalating privileges vs. gaining an access to a local account from a remote machine), both U2R and R2L classes are mainly performed via brute-force and vulnerability exploitation. Its data collection lasted nine weeks during weekdays, with a train–test split after the seventh week.

Since DARPA 1998 provides only raw packet and host log data, additional feature extraction and labeling steps are required to utilize

Table 4
Comparative table (spans multiple pages) of available public network intrusion datasets with 13 extracted properties. See Section 4 for their explanation.

Dataset	Year	Focus	Normal	Attack	Format	# Ftrs.	Anon.	Size	Duration	Net type	Compl.	Split	Labels
DARPA 1998 (Lippmann et al., 2000a)	1998	General IDS (LANs)	emul-pv	emul-pv (32) BDRVO	packet (bin) logs (host)	–	No	33.8M + 2.2M pkts	7 × 5 d + 2 × 5 d periodic	Large military	Yes	Yes	Yes
GureKDDcup (Perona et al., 2008)	1998	Payload analysis IDS	emul-pv	emul-pv (27) BRVO	nethost	47	No	178.8k recs[a]	7 × 5 d + 2 × 5 d periodic	Large military	Yes	No	Yes
KDD Cup 1999 (Stolfo et al., 2000)	1998	General IDS (LANs)	emul-pv	emul-pv (38) BDRVO	nethost	42	No	4.9M + 311k recs	7 × 5 d + 2 × 5 d periodic	Large military	Yes	Yes	Yes
NSL-KDD (Tavallaee et al., 2009)	1998	General IDS (LANs)	emul-pv	emul-pv (32) BDRVO	nethost	43	No	126k + 22.5k recs	7 × 5 d + 2 × 5 d periodic	Large military	Yes	Yes	Yes
DARPA 1999 (Lippmann et al., 2000b)	1999	General IDS (LANs)	emul-pv	emul-pv (58) BDRVO	packet (bin) logs (host)	–	No	57.5M + 14.6M pkts	3 × 5 + 2 × 5 d periodic	Large military	Yes	Yes	Yes
LLDOS (Haines et al., 2001)	2000	Infiltration & DDoS attacks	emul-pv	emul-pv (3) DRV	packet (bin) logs (host)	–	No	1.044M + 585k pkts	3.3 h + 1.7 h cont.	Large military	Yes	Yes	Indir.
MAWILab (Fontugne et al., 2010)	2001– Now	Network anomalies	real	real DR?	packet (bin)	–	IPs payload	Dozens of millions pkts/day	15 m/day[b] periodic	Large internet backbone	Yes	No	Yes
CAIDA DDoS Attack 2007 (CAIDA, 2010)	2007	DDoS attacks	No	real (1) D	packet (bin)	–	IPs payload	46M pkts	1 h 6 m cont.	n. s	No	No	Yes
Twente (Sperotto et al., 2009)	2008	NetFlow General NIDS	No	real (1) BR?	flows (uni)	13	IPs	14.2M flows	6 d 10 h cont.	University honeypot	No	No	Yes
ASNM-CDX (Homoliak et al., 2020)	2009	Buffer overflow	emul-p	emul-p (1) V	flows (bi)	194	No	5.7k flows	n. s.	Small	Yes	No	Yes
CDX (Sangster et al., 2009)	2009	General NIDS (LANs)	emul-p	emul-p DRO?	packet (bin) logs (both)	–	No	24.4M pkts	4 × 8 h periodic	Small	Yes	No	Indir.
ISOT Botnet (Saad et al., 2011)	2004–2010	Botnet traffic	emul-p	emul-p (2) CO	packet (bin)	–	No	162M pkts	n. s.	Small	Yes	No	Yes
ISCX-IDS-2012 (Shiravi et al., 2012)	2010	General NIDS (LANs)	emul-p	emul-p (5) BCDV	packet (bin) flows (bi)	21	No	120M pkts 2.5M flows	7 d cont.	Small	Yes	No	Yes
HogZilla (HogZilla, 2015)	2010 2011	Botnet traffic	emul-p	real (8) CDRO	flows (bi)	204	No	990k flows	n. s.	Mixed	No	No	Yes
CTU-13 (García et al., 2014)	2011	Botnet traffic	real	real (8) CDRO?	packet (bin) flows (uni) flows (bi)	12 (uni) 15 (bi)	payload	856M pkts 81.4M uni-f 20.6M bi-f	5 d 23 h discont.	Large academic	Yes	No	Yes
Booters (Santanna et al., 2015)	2013	UDP amplifi- cation DDoS	No	real (1) D	packet (bin)	–	IPs	119.6M pkts	2 d cont.	Mixed[c]	No	No	Yes
Botnet (Biglar Beigi et al., 2014)	2004–2014	Botnet traffic	emul-p	real (8) & emul-p (8) CDRO	packet (bin)	–	No	9.4M + 5.1M pkts	n. s.	Mixed	Yes	Yes	Yes
SSHCure (Hofstede et al., 2014)	2013 2014	SSH brute-force	real	real (1) B	flows (uni) logs (host)	10	IPs	360.87M flows	2 × 1 mo periodic	Honeypot & Large academic	Yes	No	Indir.
ASNM-TUN (Homoliak et al., 2020)	2015	Buffer overflow	real & emul-v	emul-v (2) VO	flows (bi)	194	IPs payload	394 flows	n. s.	Small & academic	Yes	No	Yes

(*continued on next page*)

Table 4 (*continued*).

Dataset	Year	Focus	Normal	Attack	Format	# Ftrs.	Anon.	Size	Duration	Net type	Compl.	Split	Labels
AWID2 (Kolias et al., 2016)	2015	Wireless NIDS	emul-p	emul-p (15) DRO	packet (txt)	156	No	37.8M + 4.57M recs	96 h + 12 h cont.	Small wireless	Yes	Yes	Yes
Kyoto 2006 (Song et al., 2011)	2006–2015	General IDS	real	real DRV?	other	24	IPs	806.1M flows	9 y 2 mo span discont.	Honeypot & legit server	No	No	Yes
UNSW-NB15 (Moustafa and Slay, 2015)	2015	General IDS (LANs)	emul-pv	emul-pv (9) DRVO	packet (bin) flows (bi) logs (net)	49	No	187M pkts 2.54M flows	16 h + 15 h cont.	Small	Yes	Yes	Yes
DDoS 2016 (Alkasassbeh et al., 2016)	2016	DDoS attacks	emul-v	emul-v (4) D	packet (txt)	28	IPs	2.16M pkts	n. s.	n. s.	n. s.	No	Yes
Kent 2016 (Kent, 2016)	2016	Anomaly- based IDS	real	emul-p (–) ?	flows (uni) logs (both)	9	IPs time names	130M flows 40.8M net logs	29 d (net) 57 d (all) cont.	Large enterprise	Yes	No	Indir.
NDSec-1 (Beer et al., 2017)	2016	Attacks for salting	emul-v	emul-v (12) BCDPVO	packet (bin) logs (host)	–	No	3.77M pkts	4 h 11 m discont.	Small	Yes	No	Yes
NGIDS-DS (Haider et al., 2017)	2016	General IDS (LANs)	emul-pv	emul-pv (7) DRVO	packet (bin) logs (host)	–	No	1.09M pkts	5 d 21 h cont.	Small	Yes	No	Indir.
UGR'16 (Maciá-Fernández et al., 2018)	2016	Cyclostation- ary NetFlow NIDS	real	real (4) & emul-pv (3) CDRO?	flows (uni)	12	IPs	13,000M + 3900M flows	3 mo 9 d + 1 mo 3 d cont.	Large ISP	Yes	Yes	Yes
Unified Host and Network (Turcotte et al., 2019)	2016	Anomaly- based IDS	real	n.s. ?	flows (bi) logs (host)	11	IPs, time names	17,883M flows	3 mo cont.	Large enterprise	Yes	No	No
2017-SUEE8 (Lukaseder, 2017)	2017	HTTP(S) Slow DoS	real	emul-pv (3) D?	packet (bin)	–	IPs, MACs	19.3M pkts	8 d 7 h cont.	Large academic	Yes	No	Yes
CIC-IDS2017 (Sharafaldin et al., 2018b)	2017	General NIDS (LANs)	emul-p	emul-p (14) BCDRV	packet (bin) flows (bi)	79	No	56.4M pkts 3.1M flows	5 × 8 h periodic	Small	Yes	No	Yes
CIC-IDS2017 Impr. (Liu et al., 2022; Engelen et al., 2021)	2017	General NIDS (LANs)	emul-p	emul-p (15) BCDRV	flows (bi)	91	No	2.1M flows	5 × 8 h periodic	Small	Yes	No	Yes
CIDDS-001 (Ring et al., 2017c)	2017	NetFlow General NIDS	real & emul-v	emul-pv (4) BDR?	flows (uni)	16	IPs	32M flows	28 d 6 h cont.	Small enterprise	Yes	No	Yes
CIDDS-002 (Ring et al., 2017b)	2017	NetFlow Port scans	emul-v	emul-v (1) R	flows (uni)	16	IPs	16.2M flows	13 d 19 h discont.	Small enterprise	Yes	No	Yes
ISOT HTTP Botnet (Alenazi et al., 2017)	2017	Botnet traffic	emul-v	emul-v (9) C	packet (bin)	–	No	10.6M pkts	54 d 17 h discont.	Small	Yes	No	Yes
IoT host-based ID dataset (Bezerra et al., 2018)	2017	IoT botnet activity	emul-p	real (1) CDV	packet (bin) flows (bi) logs (host)	47	Yes[d]	9.34M pkts 1.72M flows	11 h 55 m discont.	Small	No	No	Yes
LYCOS-IDS2017 (Rosay et al., 2022)	2017	General NIDS (LANs)	emul-p	emul-p (13) BCDRV	flows (bi)	83	No	1.8M flows (661k+220k)	5 × 8 h periodic	Small	Yes	Yes	Yes
TrabID (Viegas et al., 2017)	2017	Network anomalies	emul-p	emul-p (16) DR	packet (bin) packet (txt)	44	No	469.4M pkts	8 h discont.	Medium	Yes	Yes	Yes

(*continued on next page*)

Table 4 (*continued*).

Dataset	Year	Focus	Normal	Attack	Format	# Ftrs.	Anon.	Size	Duration	Net type	Compl.	Split	Labels
ISOT-CID (Aldribi et al., 2020)	2016 2018	Cloud IDS	real & emul-pv	real & emul-pv (18) BDRVO?	packet (bin) logs (host)	–	n.s.[e]	24.5M + 12.4M pkts	5 d + 6 d discont.	Medium enterprise cloud	Yes	Yes	Yes
Kitsune (Mirsky et al., 2018)	2017 2018	IoT NIDS	emul-p	emul-p (9) BCDRO	packet (bin) other	115	No	2.77M pkts	6 h 50 m discont.	Small IoT	Yes	No	Yes
NetML (Barut et al., 2020)	2013 2017 2018	General IDS	real	real (20) CDO?	flows (bi)	63	IPs	484k flows	n. s. discont.[f]	Large academic	Yes	Yes	Yes
NF-UQ-NIDS/ NF-UQ-NIDS-v2 (Sarhan et al., 2021, 2022b)	2015 2017 2018	NetFlow General NIDS	emul-pv	emul-pv (20) BCDRVO	flows (bi)	15/45	No	11.99M/ 79.99M flows	58 d 12 h discont.	mixed (3 small networks)	Yes	No	Yes
ASNM-NPBO (Homoliak et al., 2020)	2018	Buffer overflow	real & emul-v	emul-v (2) VO?	flows (bi)	194	IPs payload	11.4k flows	n. s.	Small & academic	Yes	No	Yes
Bot-IoT (Koroniotis et al., 2019)	2018	IoT NIDS	emul-v	emul-v (10) DRO	packet (bin) flows (bi)	47	No	549.8M pkts 73.4M flows (2.9M + 0.7M)	20 d cont.	Small IoT	Yes	Yes	Yes
CSE-CIC-IDS2018 (Sharafaldin et al., 2018b)	2018	General NIDS	emul-v	emul-v (14) BCDRV	packet (bin) flows (bi) logs (host)	84	No	1,18M pkts 16.2M flows	10 d discont.	Large	Yes	No	Yes
CSE-CIC-IDS 2018 Impr. (Liu et al., 2022)	2018	General NIDS	emul-v	emul-v (14) BCDRV	flows (bi)	91	No	63.2M flows	10 d discont.	Large	Yes	No	Yes
N-BaIoT (Meidan et al., 2018)	2018	IoT botnet activity	emul-p	emul-p (10) DRO	other	115	No	7.06M pkts	n. s.	Small IoT	Yes	No	Yes
IoT-23 (Garcia et al., 2020)	2018 2019	IoT botnet activity	real	real (9) CDRVO?	packet (bin) flows (bi)	23	No	814M pkts 325M flows	21 d discont.	Small IoT	Yes	No	Yes
CIC-DDoS2019 (Sharafaldin et al., 2019)	2019	Flooding DDoS	emul-p	emul-p (12) D	packet (bin) flows (bi)	87	No	251M + 61M pkts, 50M + 20.3M flows	7 h + 8 h cont.	Small	Yes	Yes	Yes
CUPID (Lawrence et al., 2022)	2019	General NIDS (LANs)	emul-pv	emul-pv (1) BRVO	packet (bin) flows (bi)	84	No	50M pkts 1.5M flows	7 d 16 h discont.	Small	Yes	No	Yes
GTCS (Mahfouz et al., 2020)	2019	General NIDS (LANs)	emul-v	emul-v (4) BCDV	flows (bi)	84	No	517k flows	9 d cont.	Small	Yes	No	Yes
IoT network intrusion (Kang et al., 2019)	2019	IoT NIDS	emul-p	emul-p (9) BDRO	packet (bin)	–	No	2.99M pkts	2 h 1 m discont.	Small IoT	Yes	No	Yes
IoTID20 (Ullah and Mahmoud, 2020)	2019	IoT NIDS	emul-p	emul-p (8) BDRO	flows (bi)	86	No	625.7k flows	2 h 13 m discont.	Small IoT	Yes	No	Yes
TON_IoT (Alsaedi et al., 2020; Moustafa, 2021)	2019	IoT and IIoT IDS	emul-pv	emul-pv (9) BDRVO	packet (bin) flows (bi) logs (host)	46	No	212M pkts 22.34M flows	6 d 12 h discont.	Small IoT	Yes	No	Yes
MedBIoT (Guerra-Manzanares et al., 2020)	2019	IoT botnet activity	emul-pv	emul-pv (3) CRV	packet (bin) packet (txt)	100	No	37.4M pkts	10 d 11 h cont.	Small IoT	Yes	No	Yes
WUSTL-IIoT-2021 (Zolanvari et al., 2021)	2019	IIoT NIDS	emul-pv	emul-pv (4) DRV	flows (bi)	49	No	1.19M flows	7 h 2 m cont.	Small IIoT	Yes	No	Yes

(*continued on next page*)

Table 4 (*continued*).

Dataset	Year	Focus	Normal	Attack	Format	# Ftrs.	Anon.	Size	Duration	Net type	Compl.	Split	Labels
LITNET-2020 (Damasevicius et al., 2020)	2019 2020	General NIDS	real	emul-pv (12) DRVO?	flows (bi)	85	IPs	39.6M flows	10 mo cont.	Large academic	Yes	No	Yes
InSDN (Elsayed et al., 2020)	2019 2020	SDN NIDS	emul-v	emul-v (7) BCDRV	packet (bin) flows (bi)	84	No	15.3M pkts 344k flows	5 d 7 h discont.	Small SDN	Yes	No	Yes
X-IIoTID (Al-Hawawreh et al., 2022)	2019 2020	IoT and IIoT IDS	emul-p	emul-p (18) BCDRVO	nethost	68	No	820.8k recs	6 d 7 h 8 m discont.	Small IoT	Yes	No	Yes
BOUN DDoS (Erhan and Anarım, 2020)	2020	DDoS attacks	real	emul-p (2) D?	packet (txt)	12	IPs time	17.38M pkts	2 × 8 m periodic	Large academic	Yes	No	Yes
CCD-INID-V1 (Liu et al., 2021b)	2020	Smart home IoT IDS	emul-p	emul-p (5) BDO	flows (bi)	88	No	91.6k flows	13 h 45 m discont.	Small IoT	n. s.	No	Yes
DAPT 2020 (Myneni et al., 2020)	2020	APT simulation	emul-pv	emul-pv (12) BRVO	packet (bin) flows (bi) logs (host)	85	No	77.1M pkts 86.7k flows	5 × 8 h periodic	Small	Yes	No	Yes
OPC UA (Pinto, 2020)	2020	CPS IDS for OPC UA	emul-p	emul-p (3) DO	flows (bi)	32	No	108k flows	2 h 3 m cont.	Small IIoT	Yes	No	Yes
SDN Dataset (Sarica and Angin, 2020)	2020	SDN NIDS	emul-v	emul-v (5) DRV	flows (bi)	33	No	27.92M + 30.2M flows	44 h + 42 h discont.	Small SDN IoT	Yes	Yes	Yes
SR-BH 2020 (Riera et al., 2022)	2020	HTTP(S) web attacks	real	real (13) BRVO?	flows (bi)	38	No	907.8k flows	12 d 10 h discont.	Honeypot	No	Yes	Yes
UKM-IDS20 (Al-Daweri et al., 2021)	2020	General NIDS (LANs)	emul-v	emul-v (7) DRVO	flows (bi)	48	IPs	10.3k + 2.6k flows	1 d + 1 d cont.	Small	Yes	Yes	Yes
VOIP Enterprise – Attack (Alvares et al., 2021)	2020	VoIP NIDS	emul-p	emul-p (6) BDRO	packet (bin)	–	No	4.48M pkts	3 h 38 m discont.	Small VoIP	Yes	No	Yes
AWID3 (Chatzoglou et al., 2021)	2021	Wireless NIDS	emul-p	emul-p (13) BCDVO	packet (bin) packet (txt)	254	No	37M pkts	2 h 20 m discont.	Small wireless	Yes	No	Yes
CyberFORCE Scenario (CyberVAN, 2021)	2021	APT simulation	emul-v	emul-v (5) CRVO	packet (bin) logs (both)	–	No	184k pkts	2 h 30 m cont.	Small	Yes	No	Indir.
Edge-IIoTset (Ferrag et al., 2022)	2021	IoT and IIoT NIDS	emul-p	emul-p (14) BDRVO	packet (bin) packet (txt)	63	No	20.9M pkts	1 d 23 h discont.	Small IoT	Yes	No	Yes
I-Sec (Serinelli, 2023)	2021	General NIDS (LANs)	emul-v	emul-v (7) DR	flows (bi)	78	IPs time	532k flows	n. s.	Small P2P	No	No	Yes
PWNJUTSU (Berady et al., 2022)	2021	APT simulation	No	emul-v BRVO	packet (bin) nethost logs (host)	160[g]	No	178.4M pkts 45.1M recs	22 × 1–15 d periodic 116 d total	Small	No	No	No
Unraveled (Myneni et al., 2023)	2021	APT simulation	emul-v	emul-v (5) BCVO	packet (bin) flows (bi) logs (both)	89	No	252M pkts 6.86M flows	40 d 13 h cont.	Small enterprise	Yes	No	Yes
USB-IDS-1 (Catillo et al., 2021a)	2021	Layer 7 web DoS	emul-p	emul-p (4) D	flows (bi)	84	No	4.5M flows	16 × 10 m & 2 d 19 h discont.	Small	Yes	Yes	Yes
LUFlow (Mills, 2022)	2020–2022	Network anomalies	real	real (2) ?	flows (bi)	16	IPs	206.56M flows	8 mo cont.	Large academic	No	No	Yes

(*continued on next page*)

Table 4 (*continued*).

Dataset	Year	Focus	Normal	Attack	Format	# Ftrs.	Anon.	Size	Duration	Net type	Compl.	Split	Labels
CIC IoT dataset 2022 (Dadkhah et al., 2022)	2021 2022	Home IoT monitoring	emul-p	emul-p (2) BD	packet (bin) other	48	No	120M pkts 466k recs	22 d 13 h discont.	Small IoT	No[h]	No	Indir.
HIKARI-2021 (Ferriyan et al., 2021)	2021 2022	Encrypted traffic NIDS	real & emul-p	emul-p (4) BRO?	packet (bin) flows (bi)	86	IPs payload	66.5M pkts 783k flows	1 d 4.5 h discont.	Small	Yes	No	Yes
UWF-Zeek Data22 (Bagui et al., 2023)	2021 2022	General NIDS (LANs)	emul-v	emul-v (2[i]) R	flows (bi)	23	No	18.56M flows	1 mo 22 h discont.	Small	Yes	No	Yes
VHS-22 (Szumelda et al., 2022)	2004–2022	General NIDS	real & emul-pv	real & emul- pv (115) BCDRVO?	flows (uni)	47	No	27.7M flows	5 d 5 h discont.	Mixed	Yes	No	Yes
AIT Log Dataset 2.0 (Landauer et al., 2023)	2022	General IDS (LANs)	emul-v	emul-v (22) BRVO	flows (bi) logs (both)	145	No	224M pkts 3.5M flows	8 × 4–6 d periodic 40 d total	Small	Yes	No	Yes
ICS-Flow (Dehlaghi-Ghadim et al., 2023)	2022	ICS NIDS	emul-v	emul-v (5) DRO	packet (bin) flows (bi)	64	No	25.2M pkts 45.7k flows	2 h 20 m cont.	Small industrial	Yes	No	Yes
OD-IDS2022 (Patel et al., 2023)	2022	General NIDS (LANs)	emul-pv	emul-pv (28) BDVO	flows (bi)	82	No	103.2k flows	1 mo discont.	Small	Yes	No	Yes
Simargl2022 (Komisarek et al., 2022)	2022	NetFlow General NIDS	real & emul-pv	emul-pv (3) CDR?	flows (bi)	31	n. s.[j]	30.1M flows	5 d 1 h discont.	Large academic	Yes	No	Yes
UNR-IDD (Das et al., 2023)	2022	Port-based SDN NIDS	emul-v	emul-v (5) DRO	other	34	No	37k recs	n. s.	Small SDN	Yes	No	Yes
CICIoT2023 (Neto et al., 2023)	2022 2023	IoT NIDS	emul-p	emul-p (33) BCDRVO	packet (bin) other	47	No	4,49M pkts 46.7M recs	6 mo 15 d discont.	Medium IoT	Yes	No	Yes & Indir.
TII-SSRC-23 (Herzalla et al., 2023)	2022 2023	General NIDS (LANs)	emul-p	emul-p (26) BCDR	packet (bin) flows (bi)	86	No	43.6M pkts 8.7M flows	1 mo 1.5 d discont.	Small	Yes	No	Yes
Appraise H2020 (Komisarek et al., 2023)	2023	NetFlow anomalies	real	emul-p (3) BDR?	flows (bi)	19	n. s.[j]	15.12M flows	2 mo 12.5 d discont.	Large enterprise	Yes	No	Yes
FLNET2023 (Kumar et al., 2023)	2023	Federated learning NIDS	emul-v	emul-v (11) DVO	packet (bin) flows (bi)	83	No	370M + 79M pkts || 5.42M + 747k flows	6 d 2 h + 11 d 4 h discont.	Large ISP	Yes	Yes	Yes
ROSIDS23 (Değirmenci et al., 2023)	2023	Robotic arm ICS NIDS	emul-p	emul-p (4) DO	packet (bin) flows (bi)	84	No	176.3M pkts 136.6k flows	10 h 6 m discont.	Small industrial	Yes	No	Yes
Westermo (Strandberg et al., 2023)	2023	ICS NIDS	emul-pv	emul-pv (6) BRO	packet (bin) flows (bi)	64	No	4.7M pkts 68.7k flows	1 h 30 m cont.	Small industrial	No	No	Yes

[a] Additional 2.76 millions records are provided, but unlabeled.
[b] The project has been ongoing for over 20 years, capturing 15 min of traffic every day. Unfortunately, only a few days within a year were labeled since 2021.
[c] Small target network attacked by a large botnet (30k unique IPs).
[d] Anonymization details are unspecified in the documentation, but sanitization was detected in the associated PCAP file.
[e] No anonymization technique was specified, but IP anonymization & payload trimming is suspected due to containing real traffic.
[f] Several merged captures from the Stratosphere Research Laboratory (2024). The total duration of a few days or weeks is suspected, but the exact value is unknown.
[g] Multiple files with different feature vectors. When merged, there are 160 different features, but the majority contains NULL values, making the data challenging to work with.
[h] Some of the data is captured from the whole network, but a big portion was injected with individual captures from specific isolated devices.
[i] Several other attack types and labels are present, but the amount of their samples is so negligible (under 20) that they cannot be practically utilized.
[j] No anonymization technique specified, but IP anonymization is suspected due to containing real traffic.

it. In order to simplify its usage, *KDD Cup 1999 (KDD'99)* (Stolfo et al., 2000) introduced 41 pre-extracted features with a label by processing DARPA 1998 raw data. Its practicality, respected organization backing, and general lack of alternatives made it a de facto standard for IDS benchmarks for years to come. Its two derivatives, *NSL-KDD* (Tavallaee et al., 2009) and *GureKDDcup* (Perona et al., 2008), further improve data quality by removing duplicate records and reducing the dataset size (NSL-KDD) or enriching records with packet payload data (GureKDDcup).

Other DARPA-family datasets—namely *DARPA 1999* (Lippmann et al., 2000b), extending the DARPA 1998, and *LLDOS* (DARPA 2000) (Haines et al., 2001), focusing on the infiltration and DoS attacks using the DARPA 1999 environment—were also published soon after. Nevertheless, because these datasets are provided only in raw format, they have never gained significant attention from the community.

Due to their popularity, DARPA 1998/1999, as well as their derived versions KDD'99 and NSL-KDD, have been extensively studied and criticized by the community. The criticism points toward questionable data validity (McHugh, 2000), the existence of artifacts (Mahoney and Chan, 2003), lack of real-world attack stealthiness (Silva et al., 2020), or inconsistencies between DARPA 1998 data and KDD'99 features indicating labeling issues or record duplication (Tobi and Duncan, 2018). Given these issues and the datasets' age (over 25 years), DARPA datasets are no longer recommended for IDS benchmarking, though many recent studies still rely on them.

5.1.2. Canadian institute for cybersecurity

Although several datasets have been published since the DARPA-family release, its first major competition for multi-purpose IDS benchmarking was *ISCX-IDS-2012* (Shiravi et al., 2012), by the Canadian Institute for Cybersecurity (CIC). Its authors formalized and systematized dataset generation using α- and β-profiles to describe intrusions and abstract distribution models for benign network traffic. The dataset was created in a small testbed network with 21 Windows workstations and several vulnerabilities, capturing seven days of traffic as raw packets and bi-directional flows with a basic feature set.

Building on the concept of profiles, Sharafaldin et al. (2018a) introduced the B-profile and M-profile systems and published *CIC-IDS2017* (Sharafaldin et al., 2018b), spanning 40 hours over five days. The dataset introduced encrypted benign traffic and new widespread intrusions like Heartbleed, SQL injection, or ARES Botnet. A similar scenario was executed a year later on a virtualized AWS environment with hundreds of hosts, resulting in the *CSE-CIC-IDS2018* (Sharafaldin et al., 2018b) dataset. A characteristic property of both datasets was the employment of CICFlowMeter (Lashkari, 2016) for feature extraction, introducing a novel feature set for network intrusion detection of approximately 80 (the exact number depends on the tool version) features suitable for machine learning algorithms. The datasets and the feature set itself have gained much attention in the community.

However, the quality of these datasets was not appropriate, as their users reported many errors, including duplicated features, duplicated packets, incorrect flow construction, incoherent timestamps, labeling issues, or the usage of deprecated tools for attack execution (Engelen et al., 2021; Liu et al., 2022; Rosay et al., 2022; Lanvin et al., 2023). In efforts to reduce the evaluation bias caused by faulty data, the researchers have repaired (Liu et al., 2022) or reimplemented (Rosay et al., 2022) the CICFlowMeter tool and rerun it on the available raw packet data, resulting in *Improved CIC-IDS2017* and *Improved CSE-CIC-IDS2018* datasets (Liu et al., 2022), as well as the dataset *LYCOS-IDS2017* (Rosay et al., 2022).

5.1.3. NetFlow datasets

The following paragraphs cover general-purpose datasets utilizing features exclusively from the NetFlow specification. While they could also be categorized under the next sections (5.1.4, 5.1.5), we discuss them separately due to their real-world relevance. In contrast to most NID datasets with custom features, which are unavailable in production networks without additional tools, NetFlow datasets rely on standardized features of the NetFlow architecture already deployed in many networks for monitoring purposes. Therefore, NetFlow-based NIDSs are considered more practical due to lower implementation costs and more realistic benchmark results.

The first NetFlow NID dataset, *Twente* (Sperotto et al., 2009) (2009), was collected over six days from a university honeypot server. Its authors identified real brute-force and scanning attempts and labeled the flows via alert correlation techniques. However, since data is collected from a honeypot, it lacks benign traffic, thus limiting its usage to signature-based systems.

A decoy server connected to the Internet was also utilized in *CIDDS-001* (Ring et al., 2017c). In addition, the authors created a virtual network to communicate with the server to emulate benign traffic and attacks. The dataset thus contains both emulated benign and attack traffic, as well as real-world traffic labeled as suspicious or unknown. The same environment without an external server was also used in *CIDDS-002* (Ring et al., 2017b), purely focused on scanning attacks.

Instead of obtaining real traffic from honeypots, *UGR-16* (Maciá-Fernández et al., 2018) captures NetFlow traces from a Tier-3 ISP network over three months. The dataset aims to provide periodic network activity patterns such as day-and-night cycles or weekdays and weekend cycles. It includes only flow-detectable attacks such as DoS, port scans, or botnet traffic, either emulated or injected from other sources. Since the capture contains real traffic, its authors used signature- and anomaly-based methods to detect and label malicious traffic, identifying UDP scanning, SSH scanning, and spam attacks.

Sarhan et al. (2021, 2022b) highlighted the shortage of datasets with real-world features and addressed it by creating practical NetFlow datasets. For this purpose, popular datasets—UNSW-NB15, Bot-IoT, ToN-IoT, and CSE-CIC-IDS2018—were regenerated using nProbe (Ntop, 2023), providing basic (15) and extended (45) feature sets. While available individually, the authors also merged them, resulting in the *NF-UQ-NIDS* and *NF-UQ-NIDS-v2* (Sarhan et al., 2022b) datasets.

The most recent NetFlow dataset, *Appraise H2020* (Komisarek et al., 2023), also intends to tackle the lack of real-world, real-time features suitable for network intrusion detection. It captures traffic from a shopping mall network and simulates DoS, reconnaissance, and brute-force attacks. However, it uses a minimalist feature set (19) while lacking details on the network environment and attack generation. The same research group released a similar NetFlow dataset, *Simargl2022* (Komisarek et al., 2022), simulating reconnaissance, DoS, and botnet intrusions in an academic network. Both datasets were collected using nProbe with NetFlow 9.

5.1.4. Established datasets

One of the most prominent datasets for NIDS research is *UNSW-NB15* (Moustafa and Slay, 2015) (2015). Emulated using the PerfectStorm platform,[4] it includes nine intrusion classes, such as DoS, backdoors, reconnaissance, and worms. The data is provided as raw PCAP, host logs, and 49 flow-based features. However, its documentation lacks details on data generation, leading to ambiguities in its interpretation. For instance, there is no description of the emulated vulnerabilities, types of attacks, or their parameters. Recently, issues with mislabeling, simulation artifacts, and lack of diversity have also been discovered (Flood et al., 2024). For this reason, conclusions drawn by evaluations using this dataset might be less sound. PerfectStorm was also used to generate NGIDS-DS (Haider et al., 2017), which also shares these documentation issues. In addition, it only provides raw data labeled indirectly via host logs, further limiting its usability.

[4] PerfectStorm is an application and security test platform. Homepage: https://www.keysight.com/br/pt/products/network-test/network-test-hardware/perfectstorm.html.

Other notable datasets include *GTCS* (Mahfouz et al., 2020) and *LITNET-2020* (Damasevicius et al., 2020). GTCS, generated in a simple virtualized topology for over a week, includes four attack scenarios: botnet, brute-force, DDoS, and infiltration, along with legitimate traffic from Ostinato.[5] Its 84 features were extracted using CICFlowMeter. In contrast to small-scale datasets, LITNET-2020 was captured from a large academic network over 10 months. Part of its flow-based features is based on the NetFlow protocol, but a few additional were calculated via custom scripts. The authors emulated 12 types of intrusions, but similar to previous datasets, they provide no information about used tools and their parameters. Datasets *CDX* (Sangster et al., 2009), *Kyoto 2006+* (Song et al., 2011), and *NetML* (Barut et al., 2020) have also been around for a while, but their labeling or feature sets are somewhat limited, thus not receiving much attention from the community.

5.1.5. New datasets

In recent years, several new general-purpose datasets have been released. Due to their novelty, they are expected to offer the newest benign traffic patterns and current intrusions. Therefore, the results obtained using them as benchmarks should be closer to reality. Since they were released only recently, the community has not yet thoroughly analyzed them and "decided" on their popularity, as they are not yet commonly referenced in recent scientific publications. For this reason, a separate paragraph is provided with an introduction and a brief description of each newly-released dataset.

The *CUPID* dataset (Lawrence et al., 2022) (2019) emulates a small physical network with several virtualized systems. Its objective is to provide both scripted and human-generated traffic produced by professional penetration testers, allowing researchers to investigate their differences. The dataset contains raw packets and pre-extracted traffic features via CICFlowMeter. Unfortunately, it provides only binary labels (benign/attack), thus lacking those of specific intrusions.

UKM-IDS20 (Al-Daweri et al., 2021) (2020) authors emulated two small office networks within the Hyper-V virtualization platform over two weeks. The first week is attack-free, capturing two individuals accessing the network. During the second week, DoS, scanning, exploits, and ARP poisoning attacks were introduced. The dataset contains 12.9 thousand flows of 48 features extracted by a custom algorithm.

I-Sec (Serinelli, 2023) (2021) was captured within a Peer-to-Peer (P2P) topology emulated in VirtualBox, featuring DoS and scanning attacks performed via `hping3` and `nmap` tools. The traffic features were extracted using CICFlowMeter. However, due to a simplistic topology and lack of a strategy for benign traffic generation, its value for reliable benchmarks remains questionable.

AIT Log Dataset 2.0 (AIT LDS2.0) (Landauer et al., 2023) (2022) collects log data from various systems and services while simulating multi-step attacks. Such attacks span multiple stages of a cyber kill chain (Hutchins et al., 2011), involving diverse activities like scans, vulnerability exploitation, and data exfiltration. Normal behavior is emulated via automation tools, implementing formally described state machines. The data were captured in a configurable virtual testbed across eight scenarios. Despite focusing on logs, the later released AIT NetFlow dataset is suitable for NIDSs as well. A related work, the *Kyoushi Log dataset* (Landauer et al., 2021), is a subset of AIT LDS2.0.

OD-IDS2022 (Patel et al., 2023) (2022) offers an emulation of 28 modern attacks in a virtualized network. The environment contains external and internal attacker networks targeting an Ubuntu Apache HTTP server. The features were extracted using CICFlowMeter. However, details of the environment and normal traffic profiles are lacking.

UWF-ZeekData22 (Bagui et al., 2023) (2022) contains traces from a cyber wargaming course at the University of West Florida. During the course, red (offensive) and blue (defensive) teams were tasked to attack/defend the parts of the university network. The traffic features were extracted by Zeek and labeled based on the MITRE ATT&CK (MITRE, 2024) phases using participants' logs. However, 99.9999% of malicious traffic involves reconnaissance and discovery, leaving too few samples for other phases. Therefore, the dataset consists mostly of scanning traffic, limiting its use for systems focusing on other intrusions.

VHS-22 (Szumelda et al., 2022) ("Very Heterogenous Set of Network Traffic Data", 2004–2022) combines five datasets—ISOT Botnet, CTU-13, Booters, CICIDS-2017, and Malware Traffic Analysis (MTA) project captures (Duncan, 2024). Their traffic was replayed, and timestamps reset on 2022-01-01, thus resembling a single capture (including gaps) with 115 classes. The data is provided as unidirectional flows extracted via a custom network probe. Its high diversity will likely prove challenging for some ML methods, as anomaly detectors might struggle to establish a sense of a traffic normality profile.

FLNET2023 (Kumar et al., 2023) (2023) emulates a complex network with ten routers while collecting traffic on each to facilitate federated learning (FL). In contrast to traditional ML algorithms, which suppose that all data is available at one centralized location during training, federated learning aims to train the model in a decentralized, privacy-preserving manner (Zhang et al., 2021). If datasets are not created with FL in mind, different data locations need to be simulated, often leading to an experimental bias. In FLNET2023, the authors mimicked a real-world network and emulated slow DoS, volumetric DDoS, web, and infiltration attacks, extracting features via CICFlowMeter and adding an extra location feature for FL purposes.

Finally, *TII-SSRC-23* (Herzalla et al., 2023) (2023) promotes data diversity by providing eight distinct traffic types. They are divided into six benign (network use cases like audio or video streaming, text, and background) and 26 malicious subtypes (brute-force, DoS, information gathering, and Mirai botnet). The data are provided as raw PCAP and bidirectional flows produced by CICFlowMeter. However, the dataset is heavily imbalanced, with benign traffic comprising only 0.015% (1300 out of 8.7 million) flows, posing challenges for its utilization.

5.2. Special-purpose datasets

In contrast to general-purpose datasets, special-purpose datasets focus on specific aspects of intrusion detection—a single intrusion type (intrusion-specific) with multiple variants and parameters or intrusions within a particular network environment (environment-specific), such as IoT or industrial control networks. This subsection introduces existing special-purpose datasets within the NID domain.

5.2.1. Intrusion-specific

DoS/DDoS: Denial of Service (DoS) and Distributed DoS (DDoS) are the among the most prevalent intrusion-specific datasets. These attacks have been around since the beginning of the Internet and remain a major threat to this day (Mittal et al., 2023). The first DoS dataset, *LLDOS* (Haines et al., 2001) (2000), was published by DARPA. Valuable real-world samples were published in *CAIDA DDoS Attack 2007* (CAIDA, 2010) and *Booters* (Santanna et al., 2015) datasets.

Seeking to address challenges in obtaining real-world traffic, *DDoS 2016* (Alkasassbeh et al., 2016) was emulated within a virtualized environment, providing selected per-packet features. Similar features are also present in *Boğaziçi University DDoS Dataset* (Erhan and Anarım, 2020), mixing emulated DDoS attacks with real campus traffic. *CIC-DDoS2019* (Sharafaldin et al., 2019) emulates diverse flooding attacks in raw packets and CICFlowMeter features. In contrast to flooding attacks, the *2017-SUEE-data-set* (Lukaseder, 2017) and *USB-IDS-1* (Catillo et al., 2021a) focus on layer 7 slow DoS, with the latter evaluating its effectiveness against in-built webserver defensive modules.

Botnet: Botnet detection is another crucial branch of NID research. The first dataset, *ISOT Botnet* (Saad et al., 2011) (2004–2010), mixes Storm and Walowdac botnet captures with non-malicious traffic from

[5] Ostinato is a traffic generation tool primarily used for network testing and throughput measurements. Homepage: https://ostinato.org/.

the Ericsson Research Lab in Hungary. A comprehensive set of 13 real-world botnet scenarios is provided in *CTU-13* (García et al., 2014) by running unrestricted malware in a sandbox environment within an academic network. Selected CTU-13 traces and legitimate ISCX-IDS-2012 traffic were combined into the *HogZilla* (HogZilla, 2015) dataset with custom flow-based features. ISCX-IDS-2012 and CTU-13 data were also used with the ISOT Botnet and merged into the *Botnet* (Biglar Beigi et al., 2014) dataset. In contrast to analyzing entire network traffic, *ISOT HTTP Botnet* (Alenazi et al., 2017) focuses on detecting botnets only via DNS data. Botnet detection is also closely linked to IoT, as IoT devices were historically poorly secured, making them easy targets for compromise and subsequent participation in botnets. We cover them in the environment-specific datasets subsection.

Anomaly detection: Instead of identifying specific intrusions, anomaly detectors learn patterns of regular communication and then detect deviations from them. While any dataset can be used to benchmark anomaly-based systems, some were specifically designed for this purpose. *Comprehensive, Multi-Source Cyber-Security Events* (known as Kent (2016)) and *Unified Host and Network* (Turcotte et al., 2019) datasets, collected from the Los Alamos National Laboratory enterprise network, provide flow and end-host log data with a base minimalistic feature set. In addition, heavy anonymization and the absence of direct labeling significantly limit their usability.

MaWILab (Fontugne et al., 2010), the longest-running intrusion detection capture (since 2001), utilizes the MAWI Working Group Traffic Archive (MAWI Working Group, 2024) for daily 15 min backbone traces. These traces are automatically labeled via combined anomaly detectors. However, the project's inactivity since 2015 has likely rendered these detectors outdated, hurting the ground truth accuracy. Furthermore, although the MAWI Archive continues to provide daily traces, MAWILab includes limited recent data.

Automatic labeling, this time based on external third-party Cyber Threat Intelligence, was also utilized by *LuFlow* (Mills, 2022). It provides flow-based traffic from the Lancaster University network, labeling its data as benign, malicious, or outlier. The *TrabID* dataset (Viegas et al., 2017) aims to realistically evaluate NIDSs' adaptability by dividing the data into three types likely encountered in operational networks: known, similar, and unknown. Unfortunately, its environment details were never released.

APT: Conventional IDS datasets typically focus on a certain set of attacks executed in a non-systematic way. In contrast, Advanced Persistent Threat (APT) datasets simulate real-world cyber incidents, which typically follow a well-defined sequence of steps, as outlined in the MITRE ATT&CK Matrix (MITRE, 2024). APT datasets aim to replicate realistic APT behavior, typically starting with reconnaissance and progressing toward attacker's objectives like data exfiltration, while often trying to keep such activities stealthy.

DAPT 2020 (Myneni et al., 2020) captures all APT stages performed by a red team (experienced cyber-attackers) in a virtualized environment. Realism was further increased by heavily unbalancing the traffic and including stealthy attacks, causing baseline semi-supervised detectors to perform poorly. The *Unraveled* (Myneni et al., 2023) dataset expands on this work with a more complex network, longer duration, defender's responses, attack tracks covering, and different attacker types. Professional red teamers were also used in the *PWNJUTSU* (Berady et al., 2022) dataset, which is, unfortunately, unlabeled. Another APT activity, spanning 2.5 h, is also simulated by the *CyberFORCE Scenario* (CyberVAN, 2021). Unfortunately, its data are labeled indirectly, and the corresponding documentation lacks many details.

Others: In addition to the mentioned datasets, several others focused on specific intrusions are also available. Homoliak et al. (2020) published *ANSM-CDX*, *ANSM-TUN*, and *ANSM-NPBO* for detecting buffer overflow attacks using custom flow-based features. The *SSHCure* (Hofstede et al., 2014) dataset captures real-world SSH brute-force attempts via a university honeypot. *SR-BH 2020* (Riera et al., 2022) offers flow-based data for evaluating HTTP(S) multi-label intrusion detection. In contrast to standalone datasets, *NDSec-1* (Beer et al., 2017) is designed for salting by injecting its traffic into other datasets. Finally, *HIKARI-2021* (Ferriyan et al., 2021) promotes the usage of encrypted traffic for NID purposes by employing application-layer attacks via HTTPS, extracting its features using CICFlowMeter and Zeek.

5.2.2. Environment-specific

Unlike datasets collected from traditional computer networks, the following paragraphs focus on those originating from specialized environments like Software Defined Networking (SDNs) or the Internet of Things (IoT). As a result, they produce unique benign traffic patterns and are susceptible to various environment-specific attacks, requiring tailored intrusion detection systems.

IoT (botnet): The Internet of Things (IoT) is the most prevalent specialized environment among the public NID datasets, featuring IoT devices such as sensors, cameras, or smart home appliances. Often connected to traditional networks via Wi-Fi (IEEE 802.11) (Crow et al., 1997) or IoT technologies like ZigBee (IEEE 802.15.4) (IEEE LAN/MAN Standards Committee, 2020), these devices were manufactured without cybersecurity considerations. Some are thus vulnerable to IoT malware infections and consequent participation in botnets.

Reflecting this trend, *IoT host-based ID dataset* (Bezerra et al., 2018), *N-BaIoT* (Meidan et al., 2018), *IoT-23* (Garcia et al., 2020), and *Med-BIoT* (Guerra-Manzanares et al., 2020) focus on botnet activity. Captured in controlled environments, these datasets simulate a botnet by infecting testbed hosts with popular malware families like Mirai and monitoring their behavior. In addition to botnet activity, the *IoT network intrusion dataset* (Kang et al., 2019) includes other attacks like scanning and ARP spoofing. Since this dataset provides only raw traffic, Ullah et al. processed its data using CICFlowMeter, producing the *IoTID* (Ullah and Mahmoud, 2020) dataset.

IoT (general): In addition to botnets, several datasets also explore other intrusions within IoT. *Bot-IoT* (Koroniotis et al., 2019) emulates various virtual IoT devices and performs scanning, denial of service, and information theft attacks like keylogging via Kali Linux. *TON_IoT* (Alsaedi et al., 2020; Moustafa, 2021) offers various scenarios involving physical and virtual IoT devices with a large corpus of modern attacks like ransomware. The *Kitsune Network Attack Dataset* (Mirsky et al., 2018) emulates reconnaissance, man-in-the-middle, and DoS attacks within a small IoT environment with an IP surveillance camera, including a Mirai botnet scenario.

CCD-INID-V1 (Liu et al., 2021b) captures DoS, man-the-middle, and brute-force attacks within a network of four Raspberry Pi devices, using NFStream for flow-based feature extraction. Traffic from 60 physical IoT devices is gathered within the *CIC IoT dataset 2022* (Dadkhah et al., 2022). Although primarily designed for monitoring purposes, it also includes flood and brute-force intrusion scenarios. Its data are provided as raw PCAP and custom per-packet statistical features. *CICIoT2023* (Neto et al., 2023) extends its scope to 105 devices and 33 attacks across seven categories, including the Mirai botnet. Along with the raw data, pre-extracted fixed-size packet window features are also available.

IIoT/ICS: Industrial IoT (IIoT) is an ecosystem of devices, sensors, and applications that collect, monitor, and analyze data from industrial operations (Boyes et al., 2018), oftentimes integrating Industrial Control Systems (ICS) technologies, e.g., Supervisory Control and Data Acquisition (SCADA). IIoT datasets capture the behavior of industrial networks but some also contain regular, non-industrial-specific attacks.

WUSTL-IIoT-2018 (Teixeira et al., 2018) and *WUSTL-IIOT-2021* (Zolanvari et al., 2021) present reconnaissance and exploit attacks (the latter also DoS and command injection) within a small SCADA setup with a water storage tank. Their features were extracted by Argus in a bi-flow format. *X-IIoTID* (Al-Hawawreh et al., 2022) designed connectivity- and device-agnostic emulation in a physical testbed for various attacks categorized by the ATT&CK framework. *Edge-IIoTset* (Ferrag et al., 2022) proposed a small but diverse multi-layer testbed with numerous device types. It provides a rich set of attacks, including man-in-the-middle, malware, injection, information gathering, and DoS, offering both raw and pre-extracted data.

ICS: While previous datasets mixed both IIoT and ICS traffic, *ICS-Flow* (Dehlaghi-Ghadim et al., 2023) focuses solely on the ICS environment, simulating a bottle-filling factory within a virtual testbed. Four attack types: reconnaissance, replay, DDoS, and man-in-the-middle were executed and captured via ICSFlowGenerator as custom flow-based features. Using the same setup, the *Westermo* (Strandberg et al., 2023) dataset added anomaly detection scenarios like IP address misconfiguration. Since three different capture points were used, the data is also suitable for federated learning. *ROSIDS23* (Değirmenci et al., 2023) provides attacks against the Robot Operating System (ROS), utilizing CICFlowMeter for feature extraction. Finally, the *OPC UA* (Pinto, 2020) dataset captures data from an OPC UA cyber–physical system with DoS, man-in-the-middle, and spoofing attacks.

Other: Various other datasets cover additional environments. *InSDN* (Elsayed et al., 2020), *SDN Dataset* (Sarica and Angin, 2020), and *UNR-IDD* (Das et al., 2023) present regular, as well as environment-specific attacks for Software Defined Networking (SDN). Wireless-specific attacks like disassociation and deauthentification are captured by *AWID2* (Kolias et al., 2016) and *AWID3* (Chatzoglou et al., 2021), emulating small-office-home-office (SOHO) networks. *ISOT-CID* (Aldribi et al., 2020) offers cloud-specific traces with network and end-host logs collected from three hypervisors in a production environment. Finally, the *VOIP Enterprise – Attack Dataset* (Alvares et al., 2021) focuses on Voice over IP (VoIP) attacks like VoIP-DoS.

5.3. Datasets beyond this survey

For this survey, we established explicit, non-ambiguous criteria to filter relevant works, leading to the exclusion of several well-known datasets (e.g., LBNL Pang et al., 2005 or ADFA Creech and Hu, 2013). Additionally, we excluded dozens of relevant datasets due to being publicly unavailable. As these datasets might be suitable for specific use cases, we provide a complete list of exclusions and their reasons on our GitHub repository (see Section 1).

Although network intrusion datasets backed by scientific publications (the focus of this survey) are a preferred way for benchmarking NIDSs, other sources might also provide relevant data. This subsection outlines these alternatives.

5.3.1. Data repositories

Data repositories are centralized locations to store, share, and categorize the data. For cybersecurity and intrusion detection purposes, the most relevant include *IEEE DataPort*,[6] *Kaggle*,[7] and *Zenodo*,[8] hosting many datasets from scientific publications. Kaggle and Zenodo offer open access, while data on DataPort might require a paid subscription. Additional general-purpose repositories are listed on Wikipedia,[9]

In addition to general-purpose repositories with unrestricted upload, several repositories specifically focused on network and malware traces also exist. These are typically maintained by individuals or groups, and therefore, their uploads are often restricted. We list some of the most relevant below:

- *ANT Datasets*[10] – maintained by the ANT lab at the University of Southern California. It contains various benign and malicious captures since 2006.
- *AZSecure-data*[11] – a collection of various cybersecurity datasets, including several NID datasets as well.
- *CAIDA Data*[12] – Center for Applied Internet Data Analysis offers over 100 datasets, typically raw captures from diverse locations with malicious traffic, such as DDoS.
- *Contagio*[13] – a blog with historical and contemporary malware samples, threats, observations, and traffic captures from emulated environments.
- *IMPACT*[14] – hosts over 450 datasets in 14 classes, with a few relevant to NID. However, the data is only available to researchers in the U.S. and a few approved locations.
- *Malware Traffic Analysis (MTA)*[15] – a blog sharing malware samples and their raw network behavior since 2013.
- *Security Datasets*[16] – an open-source project with dozens of malicious and benign datasets from various platforms.
- *Stratosphere Laboratory Datasets*[17] – captures malware and benign samples from controlled conditions. Hosted by a research group from Czech Technical University.
- *UMASS Trace Repository*[18] – contains various network datasets primarily for traffic classification.

5.3.2. Data lists

In contrast to repositories, data lists aggregate datasets from various locations without storing them. This part covers lists published in a non-scientific manner. For scientifically-backed ones, i.e., surveys, see Related Work (Section 8).

A list of publicly available PCAP files, primarily sourced from malware activity, is maintained by *NETRESEC*.[19] In contrast, researcher *Laurens D'Hooge* focuses on pre-processed NID datasets in a textual format.[20] Several network and malware captures are also aggregated by the *SecRepo project*.[21] The *Outlier Detection Datasets (ODDS)* project[22] collects datasets for anomaly detection, including some for cybersecurity. Finally, the GitHub repositories *Awesome Cybersecurity Datasets*[23] and *Real-CyberSecurity-Datasets*[24] compile numerous cybersecurity datasets, including those for NID.

5.4. Dataset popularity

As outlined in this paper, the Network Intrusion Detection domain has long been plagued by the lack of quality datasets. For this reason, many researchers have relied on DARPA-based datasets—KDD'99 and NSL-KDD—for over two decades. While many recent studies (Abdulganiyu et al., 2023; Yang et al., 2022; Ahmed et al., 2022; Hindy et al., 2020) still consider them the most popular, such studies tend to analyze extended time periods, sometimes dating back to 2008 (Hindy et al., 2020). While KDD'99 and NSL-KDD remain extensively used, we argue that analyzing across such long time spans fails to reflect contemporary trends in the field properly.

[6] IEEE DataPort: https://ieee-dataport.org/.
[7] Kaggle: https://www.kaggle.com/.
[8] Zenodo: https://zenodo.org/.
[9] Wikipedia datasets list: https://en.wikipedia.org/wiki/List_of_datasets_for_machine-learning_research.
[10] ANT datasets: https://ant.isi.edu/datasets/index.html.
[11] AZSecure-data: https://www.azsecure-data.org/home.html.
[12] CAIDA: https://www.caida.org/catalog/datasets/overview/.
[13] Contagio: https://contagiodump.blogspot.com/.
[14] IMPACT: https://www.impactcybertrust.org/.
[15] MTA: https://www.malware-traffic-analysis.net/.
[16] Security Datasets: https://securitydatasets.com/.
[17] Stratosphere Datasets: https://www.stratosphereips.org/.
[18] UMASS Traces: https://traces.cs.umass.edu/.
[19] NETRESEC list: https://www.netresec.com/?page=PcapFiles.
[20] L. D'Hooge data: https://www.kaggle.com/dhoogla/datasets.
[21] SecRepo: https://www.secrepo.com/.
[22] ODDS: https://odds.cs.stonybrook.edu/.
[23] Awesome Cybersecurity Datasts repository: https://github.com/shramos/Awesome-Cybersecurity-Datasets.
[24] Real-CyberSecurity-Datasets repository: https://github.com/gfek/Real-CyberSecurity-Datasets.

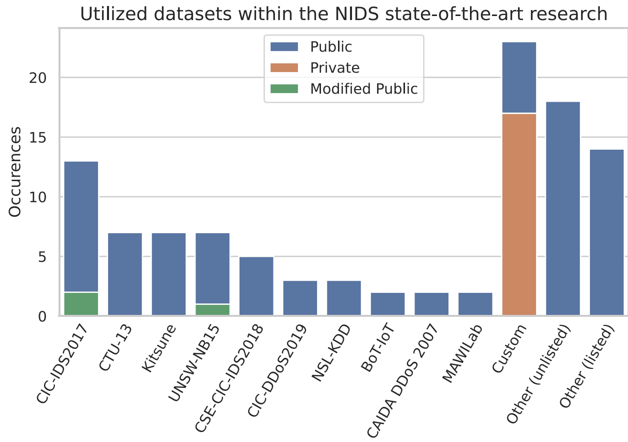


Fig. 4. Analysis of the utilized datasets from the contemporary NIDS research. Most papers used custom datasets collected specifically for the given research (category *Custom*), whereas the majority of them were not publicly shared. The category *Other (listed)* represents datasets included in the survey (Table 4) but used only once, so they were grouped. The category *Other (unlisted)* groups datasets not included in the survey.

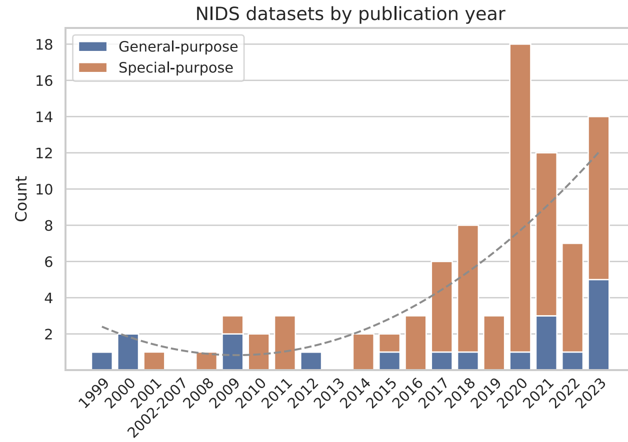


Fig. 5. The number of NID datasets by the year of publication. As shown, the number of released datasets has grown polynomially in the past years.

For this reason, we conducted our own data popularity analysis of NID papers published between 2020 and 2023 at top-tier security conferences (details in Appendix A.2). In total, we identified 45 relevant papers and summarized their utilized datasets in Fig. 4. Most papers evaluate using more than one dataset (often three or four), causing the total sum to be higher than 45. We discuss key findings below, with a complete list of papers and their analysis on our GitHub.

Custom data to get the job done: This survey shows that while many datasets exist, most current research prefers to collect its own data (category *Custom*). However, the lack of data sharing often renders the associated research non-replicable.

Due to challenges in generating realistic normal traffic, many studies collected only malicious traffic themselves and then supplemented it with normal (i.e., assumed benign) traces from other external sources like MAWI (MAWI Working Group, 2024) or CAIDA (CAIDA, 2018) to create custom mixed datasets.

Not a clear winner: Even if researchers decide to utilize public datasets, our findings indicate their usage is distributed relatively evenly. While CIC-IDS2017 emerged as the most popular, its utilization only in 29% of papers brings it far from being a "standardized" benchmark in the domain. In addition, most of its users rely on the original version with known errors, potentially undermining their research outcomes and indicating unawareness of the dataset's flaws and fixes.

Other datasets exceeding five uses include CTU-13, Kitsune, and UNSW-NB15. Datasets listed in this survey (Table 4) but utilized only once were aggregated under the *Other (listed)* category, whereas *Other (unlisted)* groups datasets not included in the table. This category mainly refers to reused datasets from other papers that produced data as a research side-effect. However, it also includes datasets like *MACCDC* (NETRESEC AB, 2024), published as a part of the NETRESEC data list, and *CIC DoS 2017* (Jazi et al., 2017), unavailable at the time of writing. For more details, refer to our GitHub.

Concerning the above results, we reason that creating custom datasets (*Custom*) or relying on custom data from similar research (*Other (unlisted)*) stems from the diversity of NID tasks. Although many datasets exist, novel research often targets highly specific scenarios (e.g., attacks or environments) not covered by public datasets. This results in a significant dataset usage variety and the need for custom data.

Are DARPA-based datasets finally dead? Unlike other studies highlighting the dominance of DARPA-based datasets, our analysis did not confirm such claims—NSL-KDD was used only three times, whereas KDD'99 only once, contradicting other surveys. We explain this discrepancy in two ways:

(a) *Surveyed years*: In contrast to other surveys, our analysis focused only on recent papers (2020–2023). As many datasets have been released recently and the DARPA flaws are now better understood, newer datasets are preferred.

(b) *Surveyed venues:* While data popularity analyses in other surveys considered diverse literature sources regardless of their ranking, our analysis selected only papers from Tier 1 and 2 security conferences (CORE A*/A ranks). As their review process is known to be more strict and emphasizes novelty, presented works are expected to be evaluated more rigorously. Given the well-known flaws of DARPA data, we assume that researchers aiming for rigorous evaluation would indeed prioritize newer datasets.

5.5. Trends in NID dataset research

This subsection analyzes trends in datasets and their properties based on Table 4. We primarily focus on property changes over time, as well as differences between new datasets (published since 2020) and older ones. For this analysis, we consider the data publication date rather than its collection year.

5.5.1. Increase in number while narrowing the scope

At the dawn of the NIDS domain (1990s, 2000s), public datasets were scarce, forcing researchers to collect their own data to evaluate the experiments. While this practice is still widely utilized (Fig. 4), recent years have witnessed a surge in dataset publications. Notably, 51 out of 89 datasets included in this survey were collected since 2020, with a steady rise in publications since 2016, as shown in Fig. 5.

Along with publishing more data, the datasets are also becoming more specialized. Since 2016, special-purpose datasets have dominated the field (83% of published datasets), a significant shift from the previous balance with general-purpose ones. Among environment-specific datasets, IoT is the most popular environment, present in 15% of all datasets (20% with IIoT and industrial networks). From an attack perspective, most prevalent intrusion-specific datasets focus on botnet traffic (10%) and DoS/DDoS (9%). Additionally, since 2020, four Advanced Persistent Threat (APT) datasets have been released, emphasizing the growing significance of these threats.

5.5.2. Traffic generation strategies

Providing realistic normal and attack traffic should be a primary objective of every dataset. As discussed throughout this paper, the network traffic can be acquired through real capture, emulation, and synthetic generation—each with distinct advantages and drawbacks.

This part elaborates on these techniques and explores their applications in existing datasets.

Normal traffic: While capturing normal traffic from a production network offers the most realism, the associated privacy concerns and anonymization needs can be highly discouraging. For this reason, only 17% of datasets provide purely *real* normal traffic, while an additional 8% combine it with emulation.

Aiming to overcome the issues with real traffic, many datasets utilize emulation: 71% rely solely on it, while additional 8% include it alongside other traffic types. This trend is even stronger for new datasets—84% include emulation since 2020. While physical emulation is the most common (41% of emulated datasets), virtual emulation (32%) and their combination (27%), are also used. Notably, virtual emulation is gaining more prominence (60% of emulated datasets since 2020) due to improved computing and virtualization technologies.

In addition to the emulation environment (physical or virtual), normal traffic is also characterized by its generation method. In this regard, we observed four common patterns:

1. *Emulation based on profiles*: Pioneered by DARPA datasets and formalized by Shiravi et al. (2012), this mechanism is based on creating profiles simulating user behaviors. The profiles, implemented via automata and scripts, perform various tasks such as interacting with applications, browsing websites, or checking e-mails. This method could create realistic traffic patterns depending on profile robustness. Datasets using this method include CIC-based data (Shiravi et al., 2012; Sharafaldin et al., 2018b), CIDDS-001/-002 (Ring et al., 2017c,b), Unraveled (Myneni et al., 2023), AWID3 (Chatzoglou et al., 2021), or HIKARI (Ferriyan et al., 2021).
2. *Emulation via traffic generators*: Traffic generators and performance testers are a straightforward way to simulate background activity. However, they typically produce overly simplistic (e.g., monotonous) traffic that fails to reflect real network complexity. This makes it easier for detectors to separate such traffic from anomalies and malicious patterns, leading to an overestimation of their real capabilities. Examples employing traffic generators include FLNET2023 (Kumar et al., 2023), UNR-IDD (Das et al., 2023), and GTCS (Mahfouz et al., 2020), using hping3, IPerf, and Ostinato tools respectively. Further information and a list of 92 traffic generators are provided by Adeleke et al. (2022).
3. *Incorporating humans*: The most realistic—yet most expensive—way of traffic emulation involves humans performing regular tasks on computers or interacting with the network (e.g., using IoT devices). Consequently, this approach also requires to specify and follow pre-defined user profiles to ensure traffic diversity. Datasets with this emulated behavior include CUPID (Lawrence et al., 2022) or CICIoT2023 (Neto et al., 2023).
4. *Disregarding normal traffic*: Another way of dealing with normal traffic is to exclude it entirely. These datasets, i.e., Booters (Santanna et al., 2015), Twente (Sperotto et al., 2009), and PWNJUTSU (Berady et al., 2022), are usually suitable only for specific purposes, such as testing signature-based detection systems or need to be mixed with normal traffic from another source.

Although not commonly present in public datasets, *synthetic* traffic offers a privacy-preserving alternative for generating network data. In the past, synthetic traffic, often narrowed to a network traffic matrix modeling, was generated using simple statistical methods (Nucci et al., 2005). However, recent advancements in generative artificial intelligence (GenAI) have improved its sophistication significantly.

Ring et al. (2019a) were the first to use Generative Adversarial Networks (GANs) to generate network flows synthetically. Their method used a novel technique, IP2Vec (Ring et al., 2017a), to learn meaningful vector representations of IP addresses by transforming them into a continuous feature space. Manocchio et al. (2021) introduced FlowGAN, a model based on Manifold Guided GAN architecture to generate NetFlow v9 data with more realistic data distributions. In addition to GANs, the authors of STAN (Xu et al., 2021) proposed an autoregressive neural network with convolutional layers to capture realistic spatio-temporal network traffic characteristics.

Generated traffic, whether emulated or synthetic, should aim for a high degree of realism. Synthetic generation methods are often evaluated by statistical tests that compare the distribution of generated traffic with other data, typically from the distribution the synthetic generator was trained on. For instance, Schoen et al. (2024) demonstrated a robust validation approach using eight metrics, statistical tests, and domain knowledge to assess the realism, diversity, novelty, and compliance of the synthetically generated network flow data. As presented by Cha (2007), various other distance metrics for comparing probability density functions can also be applied.

Malicious traffic: In contrast to normal traffic, malicious traffic is generally easier to obtain. While capturing and labeling *real* attacks (e.g., Booters Santanna et al., 2015) might be challenging due to ground truth uncertainty, some types of real-world malicious behavior like botnets can be contained and captured in controlled environments (e.g., CTU-13 García et al., 2014 or IoT host-based ID dataset Bezerra et al., 2018). Based on our survey, real malicious traffic is present in 19% of datasets. However, this trend is decreasing, as only 13.7% of datasets contain real attacks since 2020.

Similar to normal traffic, *emulation* was the most popular way for generating malicious traffic, used in 85% of all datasets (80% for sole emulation without other traffic types). This trend has strengthened in recent datasets, reaching 90% (86% for sole emulation). Physical emulation remains the most popular, with virtual emulation gaining in popularity. The most frequent attacks were DoS (74% of all datasets), Reconnaissance (67%), and Other (57%), rising to 65% for datasets published since 2020 due to the increase in special-purpose datasets.

In general, emulating malicious traffic is less complex than normal traffic, so rigorous statistical tests are often omitted. Instead, both physical and virtual attacks are typically simulated using real-world tools, such as in the Kali Linux suite. However, to ensure attack realism, efforts should prioritize designing diverse attack scenarios and varying attack parameters, as in the AIT Log Dataset (Landauer et al., 2023).

Although varying attack parameters improves realism, some argue it is still insufficient for modeling real-world cyber-attacks. For this reason, researchers seek to model realistic attack processes by employing professional penetration testers to perform the attacks (often aiming to be stealthy) and capture the resulting traffic. This strategy is especially popular within APT datasets like DAPT 2020 (Myneni et al., 2020) or PWNJUTSU (Berady et al., 2022).

Synthetic attack traffic is uncommon in public datasets, yet its generation and utilization are well-studied. In general, it can be used in two ways: a) oversampling or b) injection. While all synthetic traffic must comply with network functionality (align its timing characteristics with other traffic and generate valid data—e.g., a network flow cannot contain zero packets), attacks should also account for interactions with other network activity. For instance, a successful DDoS attack would affect the timing characteristics of all simultaneous communication or introduce packet drops. For this reason, ensuring data realism with synthetic network attacks requires a delicate approach.

The first use case—oversampling—addresses the problem of large class imbalance in NID data by generating samples for underrepresented classes. Training attack detectors on balanced data is expected to enhance their performance (Liu et al., 2021a). While early NID oversampling included SMOTE or ADASYN (Cieslak et al., 2006; Hu et al., 2020; Bagui and Li, 2021b), modern approaches utilize GANs (Kumar and Sinha, 2023) or even transformer-based models (Wolf et al., 2024).

Unlike oversampling, which enlarges the sample size of existing classes, injecting synthetic attacks introduces new classes not previously present in the data. This enables the simulation of attack scenarios that were never executed on the network, offering a cost-efficient way of benchmarking NIDSs, as crafting synthetic samples is typically cheaper than emulation. The Intrusion Detection Dataset Toolkit (ID2T) (Cordero et al., 2021; Vasilomanolakis et al., 2016) allows the injection of synthetic attacks into raw PCAP files consisting of background traffic while accounting for potential attack effects like packet drops or delays. Similar tools, including FLAME (Brauckhoff et al., 2008) and MACE (Sommers et al., 2004), were also developed but are no longer maintained, thus not supporting newer attacks.

5.5.3. Popular formats and feature sets

As outlined in Section 4, the most common data distribution formats include network flows and packets. Our results show that flows are more popular, present in 60% of surveyed datasets, 87% of which are bi-directional. The trend of flow-based NID has been on the rise lately, with 76.5% of datasets published since 2020 using this format. However, some argue that packet-based approach is more suitable for real-time detection and subsequent attack mitigation due to lower latency and comparable detection capabilities (Goldschmidt and Kučera, 2024). Notably, 23% of datasets include both formats.

Most packet-based datasets are distributed as raw packets (94%), while a few extract headers (and payloads) in an ASCII-encoded textual format. In this case, their features depend on the captured packet types and contents. In contrast, flow-based datasets require careful feature selection, making feature engineering an important research focus (Sarhan et al., 2022a; Ngo et al., 2024).

As no standardized feature set for flow-based data in the domain exists, dataset authors typically engineer their own features. This causes incompatibilities and degrades generalizability. Nevertheless, CICFlowMeter (Lashkari, 2016), introduced in CIC-IDS2017, challenges this trend, as 31% of surveyed flow-based datasets used it, making it the most popular tool for feature extraction. It is also adopted by many non-CIC datasets such as GTCS (Mahfouz et al., 2020), TII-SSRC-23 (Herzalla et al., 2023), and FLNET2023 (Kumar et al., 2023), and has also been used to transform other existing datasets like ToN_IoT and Bot-IoT (Sarhan et al., 2022b).

Although a step toward feature standardization is positive in terms of generalizability and transferability, relying a single feature set can also bring downsides. Komisarek et al. (2022) criticized CICFlowMeter features as unsuitable for real-time intrusion detection. Sarhan et al. (2022b) investigated the standardization of NetFlow v9 features for NID and found that they outperform CIC-based features in attack detection.

5.5.4. Other observations

As final observations, we analyzed how the data was captured. Our results show that most datasets (60%) model small networks (under 100 hosts), rising to 72.5% for those released since 2020. This highlights a trend toward simpler, cost-efficient testbeds over larger environments.

From a temporal perspective, 29% of datasets are captured in a single continuous run, 17% are periodic, and 43% are discontinuous (i.e., gaps within the data exist), with the latter rising to 55% since 2020. Similar to simulations in small networks, discontinuous captures offer easier execution, flexible adjustments, and less strict scheduling, making them a more convenient choice for data creators.

Looking at the trends in evaluation, 84% of datasets offer a complete capture of the whole network, while the rest focus on honeypots or specific network parts. 89% of surveyed datasets are directly labeled, 8% labeled indirectly, and 3% unlabeled. Only 23% (16% since 2020) provide predefined train–test data splits. As mentioned, this design offers greater flexibility but is prone to biased evaluations and selective reporting (Arp et al., 2022; Lipton and Steinhardt, 2019).

6. Recommendations: Data selection, creation, & usage

After reviewing existing datasets, this section outlines best practices on data-related NID topics: selecting a suitable dataset (Section 6.1), creating a custom dataset (Section 6.2), and proper data handling to mitigate experimentation bias (Section 6.3).

6.1. Choosing a dataset

Using a public dataset is the most straightforward and time-efficient approach for NID experimentation. With dozens of such datasets available, the following paragraphs guide the selection of the most suitable ones for specific use cases. However, we refrain from suggesting concrete datasets, as this survey does not aim to assess data quality but rather help others to make informed decisions.

The goal guides your way: The key criterion for dataset selection is its alignment with research goals. Therefore, defining these goals first (e.g., detecting DDoS attacks on large networks) helps in analyzing properties like the dataset's focus, attacks, format, and network type to select a suitable dataset.

Have it while it is fresh: As discussed in Sections 3 and 4, the traffic patterns evolve over time, making datasets age rather quickly. Therefore, selecting up-to-date datasets with recent traffic is desirable to accurately assess system performance in real-world scenarios.

Quality estimation: While we do not assess dataset quality in this paper, we strongly recommend performing it as a part of the data selection process. Despite the previous research on the topic (Gharib et al., 2016; Haider et al., 2017; Soukup et al., 2021), no standardized formal metric for NID data quality has been established within the domain (Kenyon et al., 2020). As "quality" itself is vague, and user needs may vary, formalizing such a general metric is challenging. Therefore, we discuss data quality in a more practical sense and attempt to gauge it by analyzing (a)the realism of the data based on the rigorousness of its creation process and (b) the contents of the data itself.

A quality dataset should be realistic and error-free. Since documentation (metadata) is one of the few ways to learn about the data creation, its analysis remains a primary resource for assessing realism. Its key details include data purpose, creation process, and related limitations (more in Section 6.2.4). Given this information, datasets with robust traffic generation and realistic environments are preferred over more simplistic ones.

However, our experience shows that analyzing documentation alone is insufficient. Even with decent documentation, numerous datasets were found to contain discrepancies like incorrect entries or timestamps. For this reason, thorough data analysis (e.g., *N/A* values, incorrect timestamps, packet errors, labeling issues) is essential in addition to documentation analysis to gauge data correctness and realism.

While many studies merely point out limitations of NID data, Flood et al. (2024) offer a practical framework for auditing network datasets. Given network flows, their approach proposes various metrics to detect mislabeling, artifacts, and poor data diversity. Alongside automated tests, they also conduct several manual checks. Replicating these steps can help identify a dataset's drawbacks and assess its suitability.

Other considerations: Additional criteria, such as data popularity, might be used during dataset selection. For instance, when comparing a new proposal to existing solutions, it is important to use the same data for evaluation. However, since popular datasets are typically older, it is beneficial to include newer datasets in addition to the popular ones.

When no suitable dataset is available, another possibility is to find the data in repositories (Section 5.3.1) or to collect a custom dataset, as outlined in the following subsection.

6.2. Creating a dataset

In essence, creating a dataset involves ten steps, from defining objectives and designing scenarios to documenting and sharing the data, as shown in Fig. 6. In the following paragraphs, we elaborate on the datasets' desirable properties (Table 5) and recommendations for individual data creation steps to mitigate common dataset limitations outlined in Section 3.2.

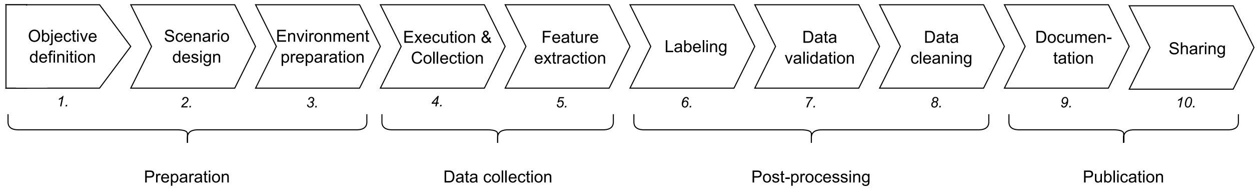


Fig. 6. The process of creating a NID dataset. The ten steps are further grouped into four phases: preparation, data collection, post-processing, and publication. Ideally, the process is linear, and the data is collected and processed in one run. However, in practice, the process is often iterative, such as re-designing and adding new scenarios or collecting and post-processing the data in multiple turns. Continually-captured datasets have to reiterate steps 2–10 periodically by design.

Table 5
Summary of desirable dataset properties. Dataset's compliance with more properties typically increases data quality. The creation step column outlines which steps during the dataset creation influence the given property. If a dataset needs to stay timely, steps 1–10 need to be periodically reiterated.

Property	Mentioned in	Creation step	Description
Correctness	Viegas et al. (2017), Małowidzki et al. (2015)	1–10	Error-free implementation of the dataset creation steps—e.g., no packet losses, correct feature extraction and labeling, documentation corresponds to the data.
Diversity	Catillo et al. (2023), Ring et al. (2019b), Maciá-Fernández et al. (2018)	2, 3	Diversity and variability of both malicious traffic (multiple launches of the same attack with differing configurations) and benign traffic (rich usage profiles).
Documentation	Landauer et al. (2023), Ring et al. (2019b), Maciá-Fernández et al. (2018)	9	Documentation (metadata) provides detailed and unambiguous information about the dataset.
Extensibility	Kenyon et al. (2020)	2, 3, 9, 10	New scenarios, attacks, and traffic capture events can be added on demand.
Label completeness	Apruzzese et al. (2022), Ring et al. (2019b), Maciá-Fernández et al. (2018), Małowidzki et al. (2015)	6	All records are labeled directly, correctly, and unambiguously.
Pre-split	Ring et al. (2019b)	2–4	Provides a predefined train–test split to support fair and objective benchmarking.
Privacy preservation	Ferriyan et al. (2021), Ring et al. (2019b), Viegas et al. (2017)	2–5, 8	The data does not compromise the privacy (reveal sensitive information) of users. Captures from real networks are properly anonymized.
Realism	Apruzzese et al. (2023a), Myneni et al. (2023), Viegas et al. (2017), Shiravi et al. (2012)	1–5, 7	The dataset contents and their characteristics reflect real-world networks. The environment is properly and completely configured. The data does not exhibit any unintended properties (e.g., artifacts) as a result of simulation.
Relevance	Landauer et al. (2023), Ring et al. (2019b, 2017b)	1–6	Dataset scenarios (e.g., network topology, attacks, normal behavior) and the data itself (format, features, labels) are appropriate and relevant for the specified dataset objectives.
Reproducibility	Catillo et al. (2021b), Landauer et al. (2023), Lewandowski (2023)	2, 3, 9, 10	The dataset contents can be regenerated on demand.
Timeliness	Apruzzese et al. (2023a), Ring et al. (2019b), Małowidzki et al. (2015)	1–3, 10	Contains up-to-date modern attacks and traffic profiles resembling the current threat landscape and the behavior of contemporary computer networks.

6.2.1. *Preparation*

The preparation phase encompasses all activities prior to the actual data collection, involving (a) defining the dataset's objective, (b) designing relevant scenarios, and (c) preparing the capture environment.

1. **Objective definition:** The first step is to define the purpose of the dataset, which then guides its entire creation process—selecting scenarios, features, and post-processing. Since many different attacks and network environments exist, creating a universal "perfect" dataset is infeasible. Instead, dataset authors should narrow down to the specific problem (e.g., attack class or adversary tactics). Its objective should cover at least three aspects: (1) malicious traffic class(es), (2) environment type and its size, and (3) time span. For instance, "*to benchmark NIDSs for Slow DoS detection on medium-sized enterprise networks over several weeks*. Formalizing a clear objective helps to determine the dataset's scope and its associated delimitations.

2. **Scenario design:** Following the specified objective, the next step is to design scenarios simulated during the execution and collection phases. Key considerations include planning on obtaining normal background traffic, malicious traffic, and their interactions. This phase also schedules network events—e.g., specific attacks, exact commands to generate them, and their relative start and end timestamps. The dataset should reflect current traffic patterns and attacks in real-world networks, making *timeliness* and *realism* top priorities. Designs are also advised to be *extensible*, enabling the addition of new scenarios and traffic patterns on demand (Viegas et al., 2017; Apruzzese et al., 2022).

As discussed in Section 6.1, documentation is crucial for assessing data realism. Nevertheless, realism is primarily shaped during scenario design and environment setup, further influenced by nearly every other action during the dataset creation. Regarding design, data authors should create scenarios aligned with the patterns observable in real-world networks. To prevent spatial bias (Pendlebury et al., 2019), datasets can also reflect realistic traffic ratios, typically by a disbalance favoring background traffic.

Further considerations should be given to *diversity*, ensuring the dataset includes diverse and variable normal and malicious traffic profiles. Insufficient variability can make the distinction between normal and malicious traffic trivial, leading to unrealistically inflated NIDS performance. *Extending capture duration* by covering day-night or workweek-weekend cycles, as in UGR'16 (Maciá-Fernández et al., 2018), is one way to enhance traffic variability. Although emulation and synthetic traffic generation methods are becoming more sophisticated, some argue that *real background traffic* must be used for realism (Ring et al., 2019b; Maciá-Fernández et al., 2018). However, it presents challenges like *privacy preservation* and establishing reliable ground truth (Landauer et al., 2023).

3. **Environment setup:** The last preparation step includes setting up a capture environment that aligns with the defined objectives and scenarios. In emulated setups, this includes configuring network equipment, end-hosts, as well as capture infrastructure, such as port mirroring or NetFlow probes, with realistic configuration and ensuring the capture will not bias the data (e.g., port mirroring introducing significant delays). Real environments utilize existing infrastructure but might add specific monitoring devices, vulnerable servers, or malicious hosts just for simulation purposes. For synthetic data, this setup refers to training or configuring a model to generate the traffic.

With the aim of maximizing data validity, some researchers suggest the datasets should be *reproducible* (Landauer et al., 2023), *extensible*, and *dynamic* (Ferriyan et al., 2021; Ring et al., 2017c; Shiravi et al., 2012), allowing their regeneration and updates with new traffic patterns and topologies. While feasible for virtual and synthetic environments, this is often impractical in physical setups. A step toward emulation reproducibility and systematization of the generation process is provided by standardized emulation platforms like MITRE Caldera (MITRE Corporation, 2024).

6.2.2. Data collection

Collecting the data comprises (a) executing scenarios and collecting the traffic and (b) extracting relevant features.

4. **Execution & collection:** After setting up the environment, the scenarios are run according to the execution plan, and relevant traffic is captured. The data format depends on the dataset goals, but *multiple data formats (and sources)*, such as packets, flows, and logs, are preferred to support benchmarking of various NIDS types (Myneni et al., 2023). However, collecting data from multiple sources presents challenges, such as clock synchronization and the need to correlate network events for accurate labeling.

5. **Feature extraction:** Feature extraction now processes the collected data into feature vectors suitable for intrusion detection. When creating a raw packet dataset or extracting features directly from NetFlow, this step is implicit during execution and collection. Otherwise, a *relevant feature set* needs to be selected in accordance with the dataset's objective, i.e., effectively describing attacks while distinguishing them from normal background traffic. Despite efforts to standardize NID feature sets (Sarhan et al., 2022b), the optimal feature set remains an open issue (Sarhan et al., 2022a). We thus encourage researchers to experiment with feature sets while ensuring they are: (a) *practical* for real-time detection (computable online), and (b) accompanied by the extraction source code and raw data to facilitate *reproducibility*.

6.2.3. Post-processing

The post-processing phase modifies the collected data to accommodate its usage as benchmarks. It involves three steps, namely data: (a) labeling, (b) validation, and (c) cleaning.

6. **Labeling:** *Complete and direct labeling* is a crucial property of NID benchmark datasets, ensuring that traffic classes are distinguishable via a simple look-up or filter upon the data. Although log files benefit certain NIDS types, they should not be used for indirect labeling due to introduced uncertainties and added effort in determining the labels. Instead, logs can supplement existing direct labels. Depending on the dataset setup, labeling can be automated or human-guided. Guerra et al. (2022) provide an overview of NID labeling techniques.

7. **Data validation**: Validating the data ensures *correctness* and *realism*—key factors in quality assurance. *Data correctness* refers to error-free capture and alignment with the designed scenarios. This involves investigating for packet losses or malformations, verifying proper attack execution and its desired effects (e.g., a DoS attack causing a service disruption), correct feature extraction (e.g., no unexpected N/A values or empty columns), the post-processing not introducing any artifacts or biases, and correct labeling. These checks are typically manual based on the data's nature and semantics, although automated tests (e.g., Flood et al. (2024)), are also viable. In addition to verifying correctness, data *realism* can be validated using statistical tests along with domain-knowledge checks, as shown by Schoen et al. (2024).

8. **Data cleaning:** In NID data creation, cleaning primarily refers to *privacy preservation*—i.e., *anonymizing* IP addresses and packet payloads to address ethical, privacy, and legal concerns for data collected from real-world networks. Additional information, like DNS queries, timestamps, or transport layer ports, may also need to be stripped. In some cases, removing entire erroneous entries might even be necessary. However, excessive anonymization can reduce realism, limiting the dataset's value (Kenyon et al., 2020). For this reason, we advocate for minimal changes to avoid the degradation of realism.

6.2.4. Publication

After collecting and processing the dataset, the final phase involves its publication. Public datasets need to be (a) documented and (b) shared, as further discussed in the following paragraphs. We also strongly advocate *specifying a static train–test evaluation split* to enable objective method comparisons and prevent introducing biases.

9. **Documentation:** *Documentation (metadata)* is one of the most important yet often neglected aspects of public datasets. Although typically finalized at the end of the process, it should record all the steps and decision-making in dataset creation. At a minimum, proper documentation should cover:

- *Goal of the dataset:* Types of NIDS is the dataset designed for. Attacks and scenarios captured within it.
- *Data generation:* A description of the procedures to generate background traffic and attacks, including specific tools, algorithms, and their parameters.
- *Environment:* Network topology, IP addressing, end-hosts, network devices and their configuration, placement of capture devices within the network, exact capture software/hardware, identification of attackers and victims.
- *Data properties:* Format of the data, size of the capture, list and description of extracted features.
- *Capture timestamps:* Exact timestamps when the data capture process started and ended. Exact timestamps of events (e.g., attacks) happening on the network.
- *Labeling:* The process of how the data was labeled.
- *Data handling:* Data transformations, anonymization, and cleaning after the data collection.
- *Data limitations:* Out-of-scope considerations. Known faults or deficiencies discovered during data validation.
- *Analysis of the captured traffic:* A brief analysis of label and protocol distributions and traffic trends (e.g., peak hours). While optional, we also recommend including benchmark detection performance with simplistic supervised and unsupervised NIDSs.

Comprehensive documentation elaborating on the above points would clarify the dataset's properties, use cases, and limitations. Despite strict space constraints in scientific papers, we encourage providing this information as additional supplementary material. Although not perfect, AWID3 dataset's paper (Chatzoglou et al., 2021) serves as an example of decent documentation.

10. **Sharing:** The final step in dataset creation is sharing its data and metadata. We recommend uploading them to multiple data storage locations, including at least one public repository (Section 5.3.1) for redundancy. In addition, publishing source code and configurations of tools for data creation (i.e., environment setup, traffic generation and collection, feature extraction, and labeling) is also highly desirable to facilitate data *reproducibility* and validation (Lewandowski, 2023). If possible, both raw packets and extracted features should be shared.

The above 10-step process produces a static dataset reflecting network behavior within a specific time period. However, as discussed in this paper, static datasets can quickly become outdated. Therefore, it is desirable to continually update datasets with new scenarios and traffic patterns by re-iterating these steps or designing specialized methodologies for collecting continual datasets, as elaborated in Section 7.

6.3. Using a dataset

After obtaining relevant data, the paper now briefly covers their proper handling. Similar to other domains, the key is to understand their semantics via exploratory data analysis and documentation, followed by standard ML procedures for data preparation like encoding and scaling (García et al., 2014). However, additional NID-specific considerations should be made. While this subsection summarizes them only briefly, works by Arp et al. (2022) and Apruzzese et al. (2023b) provide more detailed analysis.

You cannot see the future: Given the dynamic nature of computer networks (Section 3.1), benign and malicious traffic patterns can change over time. For this reason, it is crucial to respect the temporal properties of the data during evaluation. Training data timestamps must thus strictly precede evaluation (test) data to avoid *temporal data snooping*. Evaluation methods ignoring this aspect (including cross-validation) might introduce bias, leading to over-optimistic results (Arp et al., 2022; Pendlebury et al., 2019).

Some shortcuts lead to hell: Network data contain several features that might contaminate the process of data-driven learning (e.g., ML-based algorithms). These include IP addresses, port numbers, timestamps, flow IDs, or even Time to live (TTL) values. If used for training, the learner might *spuriously correlate* these artifacts with a specific activity, causing a *shortcut learning* phenomenon (Geirhos et al., 2020) and overestimating the model's performance (Arp et al., 2022; Catillo et al., 2023). For instance, D'hooge et al. (2022) demonstrated 70%–100% accuracy on popular NID datasets by considering the destination port as the sole feature.

For this reason, possible artifacts should be removed or encoded into general features (e.g., classifying IP addresses as local, global, or other) to preserve some of their informational value while preventing spurious correlations. However, encoding must be done cautiously, and the presence of spurious correlations should be verified. Identification of potential artifacts is task-specific, requiring domain knowledge and data understanding. Nevertheless, the features mentioned above ought to be problematic in most datasets and should be addressed. Heuristic identification of highly dependent features (potential artifacts) can be performed by training a model to distinguish between the attack and the background traffic using a single feature (Flood et al., 2024). Flood et al. (2024) suggest justifying the model's performance via ML explainability techniques (Nadeem et al., 2023; Jacobs et al., 2022) and connecting important features to attacks' properties.

The heaviest hammer is not always the best: As shown by numerous studies (Apruzzese et al., 2022; Zhang et al., 2022; Vinayakumar et al., 2019), even simplistic tree-based models like Random Forest or Gradient Boosting achieve near state-of-the-art performance with significantly faster execution. D'hooge et al. (2023) demonstrated that even single-rule supervised ML models perform well on most cybersecurity datasets due to the over-correlation of features with labels (Silva et al., 2022). Therefore, a general recommendation is to start with simple models before employing deep learning (Catillo et al., 2023). For practical real-time intrusion detection, it is desirable to keep models simple so their inference (i.e., attack detection) is quick, enabling efficient processing of large traffic volumes. Therefore, we suggest adopting a data-centric perspective (Zha et al., 2025), focusing on understanding and enhancing the data with simple, explainable models rather than maximizing a specific metric with complex black-box models without understanding the data.

All that glitters is not gold: As discussed in the previous paragraph, even simplistic detection models can achieve decent performance, such as accuracy above 0.99. However, this does not imply their actual usefulness. Consider an example of imbalanced traffic with a malicious-to-benign ratio of 1:100. Even if we classify all samples as benign, we receive 0.99 accuracy while not detecting any attack. On the other hand, if a detection rate was 100% with only 1% of false alarms, every second alarm produced by the model would be incorrect. Due to this phenomenon, known as the base rate fallacy (Axelsson, 1999; Alahmadi et al., 2022), and the typical NID traffic imbalance, selecting the right metrics and their correct interpretation is crucial to gauge the overall benefit of the proposed system.

Regarding specific metrics, Arp et al. (2022) suggest selecting them based on practical domain needs—primarily false alarm and detection rates for NID. To tackle the base rate fallacy, it is beneficial to incorporate metrics that account for class imbalance, namely precision, recall, precision–recall curve, as well as the Receiver Operating Characteristics (ROC) curve and the area under it (AUC-ROC). Therefore, it is crucial to utilize multiple metrics rather than relying on a single one.

Once? Not enough! Drawing conclusions based on a single evaluation run on a single dataset is generally insufficient yet relatively common in the literature (Apruzzese et al., 2023b). Ideally, models should be evaluated on multiple datasets with various splits. Conducting multiple runs (respecting temporal dependencies) allows for statistically significant comparisons using methods such as the t-test (Student, 1908) or the Mann–Whitney U-test (Mann and Whitney, 1947).

Mix them with caution: Same as with alcohol, you should also think twice before mixing different datasets. Since their data come from different networks and time periods, their traffic patterns will likely differ. Therefore, combining datasets is incorrect by design since bias would likely be introduced. This can cause automated learners to focus on dataset-specific properties (e.g., inter-packet arrival times) rather than malicious patterns as intended. Therefore, we strongly advise against mixing data from different sources unless necessary.

Despite its limitations, merging datasets in NID research is not uncommon (e.g., Fu et al. (2021)). This typically includes merging isolated attack traces with background traffic from other sources like MAWI (MAWI Working Group, 2024) or CAIDA (CAIDA, 2018). To minimize potential bias from the dataset merge, their data should be (a) collected in similar time periods, (b) from similar networks, and (c) mixed by replaying them in a local environment (Adeleke et al., 2022).

Tools for network traffic replay like *tcpreplay* (Tcpreplay, 2024) preserve packet contents while unifying time dependencies within a single environment, thus reducing time-related biases. However, merging distinct datasets can still introduce artifacts. For instance, data from different systems (e.g., Linux vs. Windows) might present differences in packet contents, such as in encryption protocols. Additionally, replaying complex attacks requiring both sides of the communication might be challenging, as most replay tools are stateless (Adeleke et al., 2022). These factors can introduce additional hidden biases during data merging. For this reason, NIDS methods should also be benchmarked on other datasets originating from a single source to prevent potential biases from skewing the results.

7. Future domain directions

Throughout this survey, we outlined the limitations of NID datasets as inherited properties from the unique intersection of cyber threat detection and computer networking domains. While Section 6 discussed best practices to tackle these limitations for immediate, personal benefits (e.g., better quality data, more sound results), this section explores long-term research and development directions to benefit the broader community.

In broad terms, future NID data research will primarily focus on the following central research question: *How to facilitate access to correct, relevant, and realistic data reflecting the current threat landscape and computer network characteristics of interest?* This question can be divided into four areas: data generation, processing, validation, and publishing, as summarized in Fig. 7 and discussed in the following subsections.

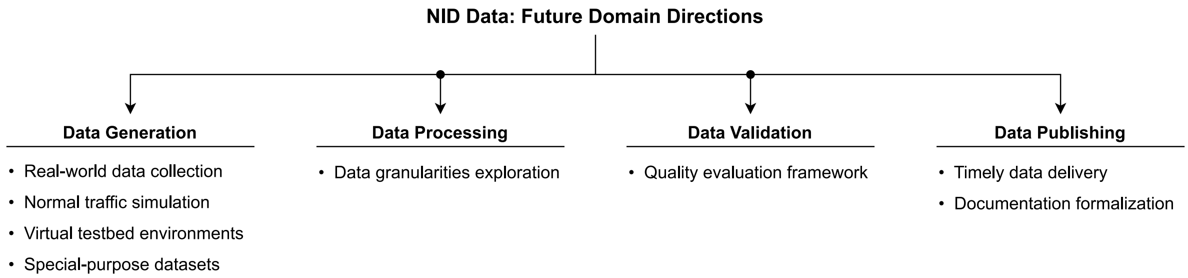


Fig. 7. Future NID data research and development directions. We divide them into areas focused on generating, processing, validating, and publishing the data.

7.1. Data generation

The research branch of data generation focuses on generating timely and realistic data. This can be achieved either by collecting real-world traces or improving traffic emulation and synthesis techniques.

Collecting timely real-world data: One of the key domain limitations is the lack of realistic real-world samples. As our survey revealed, only one in five datasets contains real traffic, with this ratio decreasing in recent years. This limitation impacts NIDS generalization, leading to overfitting to unrealistic traffic patterns and reduced performance in operational environments. For this reason, *more data with contemporary real-world network patterns are required*. While datasets CTU-13 (García et al., 2014), Appraise H2020 (Komisarek et al., 2023), and LITNET (Damasevicius et al., 2020) incorporate real traffic, they remain static and thus quickly become outdated as traffic patterns evolve.

In this matter, a key open problem is how to enable *continual real-world network data capture and annotation* while preserving privacy and ensuring label accuracy. Previous works such as Kyoto 2006+ (Song et al., 2011), MAWILab (Fontugne et al., 2010), and LuFlow (Mills, 2022) have explored this approach but are no longer updated. Currently, we are not aware of any similar project for continuous data capture providing labeled traces for NID purposes. For this reason, future research should explore *automated, privacy-preserving data collection methods from real-world environments with accurate labeling.*

Arguably, the most critical factor for reliable continual datasets is the accuracy of labels. While labeling real-world traffic cannot be fully precise, more research on *systematic continual labeling of real-world traffic*—i.e., labeling traffic with confidence in a drifting environment, is needed. While traditional labeling approaches require manual interventions, continual labeling must be automated to handle a never-ending stream of traffic. Although *alert correlation* (Kotenko et al., 2022) has improved labeling accuracy, adapting to concept drift remains an open problem (Agrahari and Singh, 2022). Since fully automated accurate labeling is difficult, integrating MLOps and human-in-the-loop methods could be a practical intermediate step (Guerra et al., 2022).

Improving simulated normal traffic: Although obtaining real-world traffic is generally considered ideal, it is often difficult or impossible due to privacy concerns, the absence of desired attack events, or operational constraints. For this reason, realistic traffic emulation remains crucial for constructing diverse and representative NID datasets. While attack traffic can be emulated via existing tools used by real-world attackers, creating realistic benign traffic remains challenging.

Existing datasets typically emulate benign traffic using profile-driven simulations (see Section 5.5.2). However, multiple studies (Layeghy et al., 2024; Engelen et al., 2022; Silva et al., 2022) have demonstrated that this traffic lacks real-world complexity, making it easily distinguishable from intrusions even with simplistic methods (D'hooge et al., 2023). This is likely due to limitations in user behavior modeling, low traffic diversity, and the lack of attack stealthiness.

To enhance traffic realism and robustness, future research should focus on *bridging the gap between simulated (emulated and synthetic) benign traffic and its real-world counterpart*. Although models like Generative Adversarial Networks (GANs) (Ring et al., 2019a) and Bayesian Networks (BNs) (Schoen et al., 2024) have already been explored, *investigating and comparing novel generative methods* like diffusion models (Yang et al., 2023) *for benign network traffic synthesis* offer promising research directions for diverse, realistic, and privacy-preserving traffic creation. Furthermore, the synthetic traffic must respect real-world constraints, making it important to *formalize domain-specific conditions for synthetic traffic validity* (e.g., network flows cannot have zero bytes).

Regarding emulation, the community would greatly benefit from a *public extensible framework for profile-based traffic generation* that could be tweaked with custom profiles to streamline the process of benign traffic creation. Methodologies *combining emulated, synthetic, and real traffic* (e.g., augmenting simulated data with real traffic) could further enhance data realism while reducing privacy risks.

Enhancing virtual testbed environments: Advances in computing, cloud, and virtualization technologies have made virtual emulation for NID data creation more feasible and popular. In addition to its cost-efficiency, virtual emulation offers many desired properties like extensibility and reproducibility, as its configuration can be fully documented and exported. Virtual environments thus help address the problem of data timeliness by allowing datasets to be updated or completely regenerated with newer traffic patterns.

Despite their benefits, reproducibility and extensibility have received little attention from the community. Most virtualized testbed datasets lack environment setup files, configuration details, or information on customizing the environment. Although AIT-LDSv2.0 (Landauer et al., 2023) emphasizes reproducibility, its environment customizability is limited. Future research should thus focus on proposing standardized *methods, frameworks, and templates for fully replicable and extensible NID data creation in virtualized and cloud environments* to enable dataset validation, modification, customization, and extension.

Creating special-purpose and scenario-based datasets: Despite the increasing number of NID datasets, many subdomains and scenarios remain underrepresented, hindering proposals' comparability and, consequently, overall research progress. For instance, all surveyed *cryptojacking detection* publications compiled their own datasets, highlighting the lack of a standardized one. Similarly, we are not aware of any public dataset for *NID in an environment with concept drift*. Although ENIDrift NIDS (Wang, 2022) includes such data, it was published as a research side-effect—thus lacking comprehensive documentation and statistical analysis—limiting its usability for benchmarking. *Long-term NID datasets incorporating cyclostationary patterns* are also needed, as UGR'16 modeling this scenario is now older.

In addition to previous scenarios, Hindy et al. (2020) and De Keersmaeker et al. (2023) pointed out that many attacks and IoT environments are not covered within public NID datasets at all. Therefore, future research should prioritize *covering specific scenarios and environments, along with ensuring robust coverage of emerging threats* to streamline NIDS research in these areas. However, without continually captured and extensible datasets is made, new datasets will remain susceptible to getting outdated due to the ever-changing nature of NID.

7.2. Data processing

Data processing focuses on handling the data after collection. While often discussed in NIDS methods research, we consider one aspect also deeply tied to the data itself.

Exploring data granularities for intrusion detection: As shown in our survey, most datasets are flow-based, while nearly half lack packet data. Flow-based NID has become the de facto standard for NID research. Similarly, many studies have explored the selection of optimal flow-based features (Mauro et al., 2021; Yin et al., 2023; Turukmane and Devendiran, 2024), but little research has been done on their effectiveness across different intrusion detection scenarios.

The focus on flow-based data has led to a relative neglect of research on alternative data granularities, such as packet-level features, time- packet-window statistics, and cross-packet contents. These formats could help detect attacks missed by flow-based methods (Umer et al., 2017). Future research should therefore investigate these alternatives and their fusion to enhance detection performance, robustness, and adaptability to evolving threats.

7.3. Data validation

Data validation evaluates collected (and processed) data to ensure its quality and correctness. It involves a single direction:

Formalizing frameworks for quality evaluation: While some metrics for NID data quality evaluation exist (Gharib et al., 2016; Haider et al., 2017), they primarily serve to compare datasets rather than assess their intrinsic quality (i.e., realism, correctness, and usability). Furthermore, their interpretation is often ambiguous due to the lack of usage instructions. Additionally, Soukup et al. (2021) employed completeness and reliability measures with an evolutionary algorithm to assess dataset quality. However, their study presents only preliminary results and lacks detailed information on the proposed framework, limiting its practical applicability. These deficiencies make evaluating and selecting suitable NID datasets difficult, leading to frequent utilization of flawed datasets that undermine research reliability.

A recent paper by Flood et al. (2024) proposed manual and automated tests to audit common limitations of NID data, such as mislabeling, artifacts, and lack of diversity. Although this represents a significant step toward data quality evaluation and the selection of suitable datasets, their heuristics focus only on a subset of typical problems, while the proposed manual analysis requires access to both flow-based and raw data.

In order to improve NID data evaluation, future research should *formalize methods and develop frameworks for evaluating NID data validity and quality*, i.e., ensure datasets are both realistic and error-free. Current data validation heavily relies on manual analysis, as seen in studies on CIC-IDS2017 and CSE-CIC-IDS2018 issues (Engelen et al., 2021; Liu et al., 2022; Lanvin et al., 2023), with Engelen et al. (2022) recommending manual analysis by domain experts before public data release. Such validations are time-consuming and dependent on the availability of experts, posing additional challenges for data creators. Therefore, research on NID data quality should also integrate expert knowledge into (semi-)automated validation methods, potentially building on the Flood et al. (2024)'s work. Standardized approaches to verify other aspects of data quality—e.g., the validity of features via domain knowledge, documentation via Large Language Models (LLMs) (Giner-Miguelez et al., 2024), or the detection of common issues—would benefit data creators to validate their datasets and assist users in selecting suitable data, enhancing NID research relevance, trustworthiness, and explainability.

7.4. Data publishing

Future directions in data publishing should focus on timely data delivery and standardization of data documentation.

Timely data delivery: Most NID datasets are uploaded to different data storage sites and later published as research papers, causing significant delivery delays for their users. However, in rapidly evolving fields like NIDS, even a few-month delay can render the data outdated. For this reason, beyond continually captured datasets discussed previously, *reducing the lag between data collection and publication* is essential.

One of the promising solutions to this issue is the development of a centralized indexing system for NID datasets, similar to the digital object identifier (DOI), as proposed by Abt and Baier (2014) and later emphasized by Ring et al. (2019b). Such a system would simplify data discovery, provide persistent availability, and reduce delivery delays. However, despite these benefits and multiple calls, we are not aware of any indexing system, data storage, or publicly maintained repository of (network) intrusion datasets.

Formalizing data documentations: While reviewing the metadata of surveyed datasets, we observed that each was different. Although documentation of all scientifically-backed datasets follows a standard paper structure, some provide detailed data descriptions while others mention the data only briefly. This leads to under-documented datasets, reducing their credibility and usability. In addition, some datasets lack scientific backing or any documentation entirely.

For this reason, future research should *formalize data documentation through guidelines, checklists, and templates to standardize NID dataset metadata*, thus enhancing data credibility and usability. Meanwhile, data creators can refer to existing informal guidelines, such as those in Section 6.2.4.

8. Related work

This section compares the presented survey with other studies discussing network intrusion detection datasets. While data-specific NID surveys (this paper) are rare, general NID surveys often mention datasets as well. Naturally, they typically lack detailed discussion on data-specific topics, yet are still a relevant reference for many readers. We thus compare our work to those focused explicitly on NID data and five general surveys with data-related sections, summarized in Table 6. Based on the related work analysis, we conclude that our paper:

1. Is the only Systematic Literature Review (SLR)-based study specifically focused on NID data,
2. Lists the highest number of public datasets, all of which are publicly available with provided download links,
3. Collects the most detailed properties (e.g., type of normal/attack traffic) for the listed datasets,
4. Performs data popularity analysis in a fully transparent way based exclusively on Tier 1 security conferences.

The closest study, and our inspiration in many aspects, is Ring et al. (2019b)'s work. This foundational work on NID data formalized data properties, compared existing datasets, and provided data-related recommendations. However, it does not include newer datasets. Our study builds on it by refining some of its properties (e.g., removing the data balance property, as NID data are naturally imbalanced) and incorporating newly released datasets. We also introduce sections on data limitations and trends to address recent findings and emphasize data quality, aiming to provide a holistic view of NID data.

The work of Kenyon et al. (2020) also overlaps with our paper, as it looks at NID data from different perspectives. The paper discusses data properties and limitations, compares 34 datasets, and closes with recommendations for NID datasets. In contrast, our study covers more datasets with detailed properties in an SLR-based approach, as well as discusses dataset popularity to better aid in selecting suitable data.

Table 6
Comparison of related work with this study. As depicted, our study covers most NID datasets (with the bonus property that all of them are publicly available), while it is the only study to perform a Systematic Literature Review (SLR) while specifically focusing on the data.

Paper	Covered until	# NID datasets	SLR-based	Data-focused	Comparative	Data limits	Popularity analysis	Data trends	Recommen-dations
Ring et al. (2019b)	2018	34	✗	✓	✓	✗	✗	✗	✓
Molina-Coronado et al. (2020)	2018	11	✗	✗	✓	✓	✓	✗	◉
Ferrag et al. (2020)	2018	23	✗	✗	✗	✗	✓	✗	✗
Hindy et al. (2020)	2018	21	✗	✓	✓	✓	✓	◉	✓
Kenyon et al. (2020)	2019	34	✗	✓	✓	✓	✗	◉	✓
Gümüşbaş et al. (2021)	2020	25	✗	◉	◉	◉	✗	✗	✗
Thakkar and Lohiya (2022)	2020	13	✓	✗	✗	✗	✗	✗	✗
Yang et al. (2022)	2021	52	✓	✗	✓	✗	✓	✗	✗
This paper	2023	89	✓	✓	✓	✓	✓	✓	✓

Symbols description: ✓: fulfilled, ◉: partially fulfilled, ✗: not fulfilled

Although Hindy et al. (2020) cover aspects similar to this paper, they mainly focus on the network threat landscape and its relation to 21 NID datasets (until 2018). The study further discusses NIDS algorithms' and datasets' popularity, threat taxonomy, IDS limitations, and recommendations for future research. In contrast, our study includes more datasets, compares them by their properties rather than by attacks, and provides a more detailed discussion of NID data-related aspects.

Yang et al. (2022) conducted an SLR on the NID domain as a whole, highlighting popular ML-based methods and datasets for benchmarking. Although not explicitly data-focused, the study presents the largest published dataset list at the time (52), compares them, and analyzes their popularity, with KDD'99 and NSL-KDD coming as the most popular. In contrast, our SLR specifically focuses on the data, listing more datasets and elaborating on data-specific topics absent in Yang et al.'s work. Other general surveys by Ferrag et al. (2020) and Gümüşbaş et al. (2021) focus on deep learning methods for network intrusion detection. Although both list about two dozen datasets, they lack a comparative analysis, data trends, and recommendations.

As mentioned in Section 2.1, this survey focuses on network intrusion datasets regardless of the environment, with the condition that the data must be captured from the network. Therefore, we exclude IDS datasets where most of the data originates from non-network sources like end-host devices or sensors, as typical for Industrial Control Systems (ICS) or Internet of Things (IoT) datasets. Although relevant for environment-specific intrusion detection, we consider such datasets out-of-scope. For this reason, we recommend exploring other, more specialized surveys for data in specific areas. For instance, Conti et al. (2021) and Koay et al. (2023) review ICS testbeds and datasets, whereas IoT-specific datasets are covered by De Keersmaeker et al. (2023) and Kaur et al. (2023).

9. Conclusions

The cybersecurity environment evolves rapidly, with new attacks and vulnerabilities emerging daily. In an effort to keep up with these trends, many datasets and data-related findings for intrusion detection were published recently. Since they are scattered across multiple places and scientific resources, this study unifies the latest findings on network intrusion detection (NID) data, provides links for data access, outlines its common issues and limitations, and offers recommendations to mitigate them. We expect the research presented in this paper to be relevant for both researchers and practitioners in the NIDS domain seeking to benchmark methods or collect their own data.

In total, this paper has *surveyed 89 NID datasets*, extracting 13 properties via manual analysis of their documentation and data. The resulting *comparative table* is a valuable resource to aid in data selection for benchmarking new NIDS proposals. Analyzing these properties, we revealed several data creation trends, such as the growing number of published datasets each year, a preference for emulation for data generation, and the rising popularity of CICFlowMeter features.

Our analysis of contemporary state-of-the-art NIDS research revealed a shift in benchmark *dataset popularity:* DARPA-based datasets are no longer the most prevalent. While CIC-IDS2017 is now frequently used, researchers still prefer to collect their own data, most of which are not publicly shared. This trend, while understandable due to the limitations of existing datasets, hinders reproducibility and validation. We emphasize the importance of following the *best practices* discussed in this paper, as well as sharing the data to improve the quality and reliability of future NIDS research.

Given the NID *domain-specific properties* and resulting *data limitations* outlined in this paper, we state that creating a perfect dataset is infeasible. We consider data timeliness as a key issue, as research relying on static datasets will inevitably lag behind ever-evolving attacks and network trends. A step toward mitigating the issue could be achieved by adopting continuously updated datasets, as discussed in Section 7.

CRediT authorship contribution statement

Patrik Goldschmidt: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization. **Daniela Chudá:** Writing – review & editing, Validation, Supervision.

Declaration of Generative AI and AI-assisted technologies in the writing process

Statement: During the preparation of this work, the authors used generative AI tools, GPT-4o and Gemini 1.5 Flash, to enhance writing in the paper—i.e. shorten its contents and improve the overall flow of the text. After using these tools, the authors reviewed and edited the content as needed and take full responsibility for the content of the publication.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Acknowledgment

This work was supported by the OZ BrAIn association.

Appendix A. Survey methodology in detail

As outlined at the paper's beginning, the survey was conducted using the Systematic Literature Review (SLR) process to achieve comprehensive, unbiased, and replicable results.

The whole SLR process, i.e., the specification of a review protocol, resource search, filtering, and results synthesis, was performed by two researchers. The first author (Ph.D. student) conducted the process under the supervision of the second author (Ph.D. supervisor), who helped in fine-tuning the review protocol and validated a random sample (10%) of the studies. This appendix elaborates on the SLR details.

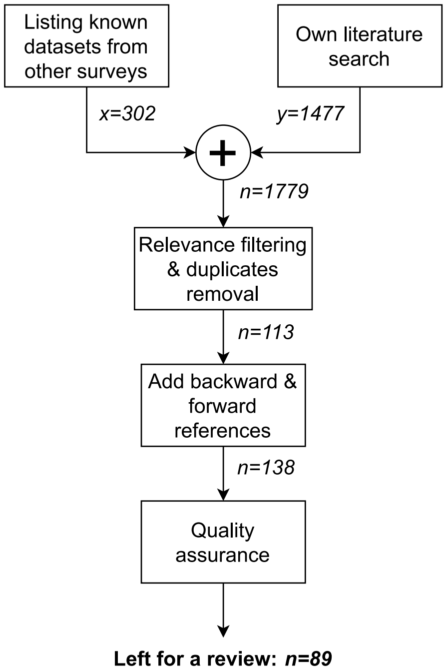


Fig. A.8. The diagram illustrates the steps of searching, filtering, and selecting relevant studies during the systematic literature review process. Specifically, we identified the objects of interest from three sources: existing survey papers, our own search via paper titles and keywords, and backward and forward reference analysis. To ensure feasibility, the reference analysis was conducted after the initial filtering. Finally, all identified sources underwent a quality assurance process to include only relevant datasets.

A.1. Data search and filtering process

In order to ensure the replicability of our research, we outline the process used to identify and filter the datasets included in the survey (Fig. A.8). The steps were as follows:

1. *Listing known datasets:* We extracted known datasets from existing network intrusion surveys listed in Table 6, resulting in 302 datasets (with duplicates).
2. *Own literature search:* We conducted a search with the following query: *network (intrusion ∪ anomaly ∪ outlier ∪ attack ∪ threat) (dataset ∪ data set)*. The publication year was limited to 2023, so papers published since 2024 are not included. Matching on article titles or author keywords, we used the following academic search engines and databases:
   - SCOPUS: 656 results
   - ACM Digital Library: 42 results
   - IEEEXplore: 179 results
   - Google Scholar: 600 results[25]

   In total, this step produced 1477 papers (with duplicates).
3. *Relevance filtering:* After collecting 302 + 1477 items from the previous steps, we excluded duplicates. Next, we selected only dataset-specific resources, as most articles from the previous step only used existing datasets. This filtering was based on title, abstract, and full-text analysis using the following criteria:

   Inclusion criteria:
   - Published until 2023 (inclusive),
   - Presents novel network data traces specifically for network intrusion or anomaly detection in the cyberthreat detection context.

   Exclusion criteria:
   - Presents an IDS method using existing data,
   - Focuses on host-based intrusions or an intrusion detection subdomain containing environment-specific data without network traces (packets, flows),
   - Purposed for network traffic classification or non-cyberthreat anomaly detection.

   After the filtering, 113 objects were left.
4. *Backward and forward reference search:* Aiming to compile a comprehensive contemporary dataset list, we also reviewed papers that cited (forward search) and were cited by (backward search) those obtained in the third step using Google Scholar. This process was limited to papers from the past five years (2018–2023). Applying the same filtering criteria as in the previous step yielded 25 additional papers, bringing the total to 138.
5. *Quality assurance:* Lastly, we aimed to ensure the quality of the selected papers. While datasets backed by scientific publications are preferable, a non-negligible portion of the data is only available via informal sources (e.g., websites), so including them is also desirable.
   Hence, we work with a mixture of scientific papers and other resources that do not follow a standard scientific paper format or provide experimental results, making common research quality assessments (Yang et al., 2021) impractical. Since we do not aim to propose new NIDS data quality metrics, our quality check is based on a simple premise: *Is the dataset useful for potential users?* Therefore, we specify:
   Quality-related exclusion criteria:
   - Was collected as auxiliary material to other research not primarily focused on data,
   - Is heavily pre-processed, thus preventing meaningful analysis and sound conclusions,
   - Is not publicly available.

Failing to reject all criteria results in the exclusion of a dataset. After their application, *89 datasets remained for a final review*.

A.2. Data popularity determination

In order to aid in the dataset selection, we also analyze dataset popularity in contemporary NIDS research. For this purpose, we focus on network intrusion/anomaly detection from Tier 1 and Tier 2 cybersecurity conferences, according to the MLSec group list.[26] In particular, we surveyed the following 12 conferences: S&P (Oakland), CCS, Security, NDSS, ACSAC, AsiaCCS, CSF, ESORICS, EuroS&P, PETS, RAID, and DIMVA, in addition to the MLSec list.

Aiming to maximize the relevance of our findings, we analyzed only papers from the past four years (2020–2023). This was done by manually reviewing conference proceedings and conducting full-text analyses to identify used datasets.

[25] Since Google Scholar returns thousands of results, we reviewed only the first 20 search pages (200 results) for all years, along with the first ten pages for 2020–2023 (400 results) to identify new datasets not covered in previous surveys. We acknowledge that this process is not entirely replicable. However, omitting it would leave over a dozen datasets undiscovered.

[26] MLSec is a research group at Technical University (TU) Berlin focused on the intersection of cybersecurity and machine learning. They list top-tier cybersecurity conferences at https://mlsec.org/topnotch/. Similar rankings are also maintained by others, e.g., Guofei Gu: https://people.engr.tamu.edu/guofei/sec_conf_stat.htm.

Data availability

Notebooks and supplementary code are available at our public GitHub repository: https://github.com/xGoldy/nid-datasets.

References

Abdulganiyu, O.H., Ait Tchakoucht, T., Saheed, Y.K., 2023. A systematic literature review for network intrusion detection system (IDS). Int. J. Inf. Secur. http://dx.doi.org/10.1007/s10207-023-00682-2.

Abt, S., Baier, H., 2014. Are we missing labels? A study of the availability of ground-truth in network security research. In: 2014 Third International Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. BADGERS, pp. 40–55. http://dx.doi.org/10.1109/BADGERS.2014.11.

Adeleke, O.A., Bastin, N., Gurkan, D., 2022. Network traffic generation: A survey and methodology. ACM Comput. Surv. 55 (2), http://dx.doi.org/10.1145/3488375.

Agrahari, S., Singh, A.K., 2022. Concept drift detection in data stream mining : A literature review. J. King Saud Univ. - Comput. Inf. Sci. 34 (10, Part B), 9523–9540. http://dx.doi.org/10.1016/j.jksuci.2021.11.006.

Ahmed, L.A.H., Hamad, Y.A.M., Abdalla, A.A.M.A., 2022. Network-based intrusion detection datasets: A survey. In: 2022 International Arab Conference on Information Technology. ACIT, pp. 1–7. http://dx.doi.org/10.1109/ACIT57182.2022.9994201.

Aitken, P., Claise, B., Trammell, B., 2013. Specification of the IP flow information export (IPFIX) protocol for the exchange of flow information. http://dx.doi.org/10.17487/RFC7011, RFC 7011.

Al-Daweri, M.S., Abdullah, S., Zainol Ariffin, K.A., 2021. An adaptive method and a new dataset, UKM-IDS20, for the network intrusion detection system. Comput. Commun. 180, 57–76. http://dx.doi.org/10.1016/j.comcom.2021.09.007, Dataset available at: https://www.kaggle.com/datasets/muatazsalam/ukm-ids20.

Al-Hawawreh, M., Sitnikova, E., Aboutorab, N., 2022. X-IIoTID: A connectivity-agnostic and device-agnostic intrusion data set for industrial internet of things. IEEE Internet Things J. 9 (5), 3962–3977. http://dx.doi.org/10.1109/JIOT.2021.3102056, Dataset available at: https://www.kaggle.com/datasets/munaalhawawreh/xiiotid-iiot-intrusion-dataset.

Alahmadi, B.A., Axon, L., Martinovic, I., 2022. 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In: 31st USENIX Security Symposium. USENIX Security 22, USENIX Association, Boston, MA, pp. 2783–2800.

Aldribi, A., Traoré, I., Moa, B., Nwamuo, O., 2020. Hypervisor-based cloud intrusion detection through online multivariate statistical change tracking. Comput. Secur. 88, 101646. http://dx.doi.org/10.1016/j.cose.2019.101646, Dataset available at: https://onlineacademiccommunity.uvic.ca/isot/2022/11/25/cloud-security-datasets/.

Alenazi, A., Traore, I., Ganame, K., Woungang, I., 2017. Holistic model for HTTP botnet detection based on DNS traffic analysis. In: Traore, I., Woungang, I., Awad, A. (Eds.), Intelligent, Secure, and Dependable Systems in Distributed and Cloud Environments. Springer International Publishing, Cham, pp. 1–18. http://dx.doi.org/10.1007/978-3-319-69155-8_1, Dataset available at: https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/botnet-and-ransomware-detection-datasets/.

Alkasassbeh, M., Al-Naymat, G., Hassanat, A.B.A., Almseidin, M., 2016. Detecting distributed denial of service attacks using data mining techniques. Int. J. Adv. Comput. Sci. Appl. 7 (1), http://dx.doi.org/10.14569/IJACSA.2016.070159, Dataset available at: https://www.researchgate.net/publication/292967044_Dataset-_Detecting_Distributed_Denial_of_Service_Attacks_Using_Data_Mining_Techniques.

Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A., 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. http://dx.doi.org/10.1109/ACCESS.2020.3022862, Dataset available at: https://research.unsw.edu.au/projects/toniot-datasets.

Alvares, C., Dinesh, D., Alvi, S., Gautam, T., Hasib, M., Raza, A., 2021. Dataset of attacks on a live enterprise VoIP network for machine learning based intrusion detection and prevention systems. Comput. Netw. 197, 108283. http://dx.doi.org/10.1016/j.comnet.2021.108283, Dataset available at: https://drive.google.com/drive/folders/1rGBsHSS3CUQa2ntQkaQfcn-tQggJcm_o.

Aouini, Z., Pekar, A., 2022. NFStream: A flexible network data analysis framework. Comput. Netw. 204, 108719. http://dx.doi.org/10.1016/j.comnet.2021.108719.

Apruzzese, G., Laskov, P., Montes de Oca, E., Mallouli, W., Brdalo Rapa, L., Grammatopoulos, A.V., Di Franco, F., 2023a. The role of machine learning in cybersecurity. Digit. Threat.: Res. Pr. 4 (1), http://dx.doi.org/10.1145/3545574.

Apruzzese, G., Laskov, P., Schneider, J., 2023b. SoK: Pragmatic assessment of machine learning for network intrusion detection. In: 2023 IEEE 8th European Symposium on Security and Privacy. Euro&P, IEEE Computer Society, Los Alamitos, CA, USA, pp. 592–614. http://dx.doi.org/10.1109/EuroSP57164.2023.00042.

Apruzzese, G., Laskov, P., Tastemirova, A., 2022. SoK: The impact of unlabelled data in cyberthreat detection. In: 2022 IEEE 7th European Symposium on Security and Privacy. EuroS&P, IEEE Computer Society, pp. 20–42. http://dx.doi.org/10.1109/EuroSP53844.2022.00010.

Ari, I., Hong, B., Miller, E., Brandt, S., Long, D., 2003. Managing flash crowds on the internet. In: 11th IEEE/ACM International Symposium on Modeling, Analysis and Simulation of Computer Telecommunications Systems, 2003. MASCOTS 2003.. pp. 246–249. http://dx.doi.org/10.1109/MASCOT.2003.1240667.

Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., Rieck, K., 2022. Dos and don'ts of machine learning in computer security. In: Butler, K.R.B., Thomas, K. (Eds.), 31st USENIX Security Symposium, USENIX Security 2022, Boston, MA, USA, August 10-12, 2022. USENIX Association, pp. 3971–3988.

Axelsson, S., 1999. The base-rate fallacy and its implications for the difficulty of intrusion detection. In: Proceedings of the 6th ACM Conference on Computer and Communications Security – CCS '99. ACM Press, Kent Ridge Digital Labs, Singapore, pp. 1–7. http://dx.doi.org/10.1145/319709.319710.

Bagui, S., Li, K., 2021a. Resampling imbalanced data for network intrusion detection datasets. J. Big Data 8 (1), 6. http://dx.doi.org/10.1186/s40537-020-00390-x.

Bagui, S., Li, K., 2021b. Resampling imbalanced data for network intrusion detection datasets. J. Big Data 8 (1), 6. http://dx.doi.org/10.1186/s40537-020-00390-x.

Bagui, S.S., Mink, D., Bagui, S.C., Ghosh, T., Plenkers, R., McElroy, T., Dulaney, S., Shabanali, S., 2023. Introducing UWF-ZeekData22: A comprehensive network traffic dataset based on the MITRE ATT&CK framework. Data 8 (1), http://dx.doi.org/10.3390/data8010018, Dataset available at: https://datasets.uwf.edu/.

Barut, O., Luo, Y., Zhang, T., Li, W., Li, P., 2020. NetML: A challenge for network traffic analytics. http://dx.doi.org/10.48550/arXiv.2004.13006, Dataset available at: https://github.com/ACANETS/NetML-Competition2020.

Beer, F., Hofer, T., Karimi, D., Bühler, U., 2017. A new attack composition for network security. In: 10. DFN-Forum Kommunikationstechnologien. Gesellschaft für Informatik e.V., Bonn, pp. 11–20, Dataset available at: https://www2.hs-fulda.de/NDSec/NDSec-1/.

Berady, A., Jaume, M., Triem Tong, V.V., Guette, G., 2022. PWNJUTSU: A dataset and a semantics-driven approach to retrace attack campaigns. IEEE Trans. Netw. Serv. Manag. 19 (4), 5252–5264. http://dx.doi.org/10.1109/TNSM.2022.3183476, Dataset available at: https://pwnjutsu.irisa.fr/.

Bezerra, V.H., da Costa, V.G.T., Martins, R.A., Junior, S.B., Miani, R.S., Zarpelão, B.B., 2018. Providing IoT host-based datasets for intrusion detection research. In: Anais do XVIII Simpósio Brasileiro de Segurança da Informação e de Sistemas Computacionais. SBC, pp. 15–28. http://dx.doi.org/10.5753/sbseg.2018.4240, Dataset available at: https://www.uel.br/grupo-pesquisa/secmq/dataset-iot-security.html.

Biglar Beigi, E., Hadian Jazi, H., Stakhanova, N., Ghorbani, A.A., 2014. Towards effective feature selection in machine learning-based botnet detection approaches. In: 2014 IEEE Conference on Communications and Network Security. pp. 247–255. http://dx.doi.org/10.1109/CNS.2014.6997492, Dataset available at: https://www.unb.ca/cic/datasets/botnet.html.

Boutaba, R., Salahuddin, M.A., Limam, N., Ayoubi, S., Shahriar, N., Estrada-Solano, F., Caicedo, O.M., 2018. A comprehensive survey on machine learning for networking: evolution, applications and research opportunities. J. Internet Serv. Appl. 9 (1), 16. http://dx.doi.org/10.1186/s13174-018-0087-2.

Boyes, H., Hallaq, B., Cunningham, J., Watson, T., 2018. The industrial internet of things (IIoT): An analysis framework. Comput. Ind. 101, 1–12. http://dx.doi.org/10.1016/j.compind.2018.04.015.

Brauckhoff, D., Wagner, A., May, M., 2008. FLAME: A flow-level anomaly modeling engine. In: Benzel, T. (Ed.), Proceedings of Workshop on Cyber Security and Test, CSET'08. USENIX Association, pp. 1–6, Tool available at: https://www.flame.ee.ethz.ch/index.html.

Buczak, A.L., Guven, E., 2016. A survey of data mining and machine learning methods for cyber security intrusion detection. IEEE Commun. Surv. Tutor. 18 (2), 1153–1176. http://dx.doi.org/10.1109/COMST.2015.2494502.

CAIDA, 2010. The CAIDA "DDoS Attack 2007" dataset. Dataset available at: https://www.caida.org/catalog/datasets/ddos-20070804_dataset/.

CAIDA, 2018. Anonymized internet traces april 2008 - january 2019. http://dx.doi.org/10.21986/CAIDA.DATA.ANONYMIZED-INTERNET-TRACES, (Accessed 12 August 2024). Available at: https://www.caida.org/catalog/datasets/passive_dataset.

Catillo, M., Del Vecchio, A., Ocone, L., Pecchia, A., Villano, U., 2021a. USB-IDS-1: a public multilayer dataset of labeled network flows for IDS evaluation. In: 2021 51st Annual IEEE/IFIP International Conference on Dependable Systems and Networks Workshops. DSN-W, pp. 1–6. http://dx.doi.org/10.1109/DSN-W52860.2021.00012, Dataset available at: https://idsdata.ding.unisannio.it.

Catillo, M., Pecchia, A., Rak, M., Villano, U., 2021b. Demystifying the role of public intrusion datasets: A replication study of DoS network traffic data. Comput. Secur. 108, 102341. http://dx.doi.org/10.1016/j.cose.2021.102341.

Catillo, M., Pecchia, A., Villano, U., 2023. Machine learning on public intrusion datasets: Academic hype or concrete advances in NIDS? In: 2023 53rd Annual IEEE/IFIP International Conference on Dependable Systems and Networks - Supplemental Volume. DSN-S, pp. 132–136. http://dx.doi.org/10.1109/DSN-S58398.2023.00038.

Cha, S.-H., 2007. Comprehensive survey on distance/similarity measures between probability density functions. Int. J. Math. Model. Methods Appl. Sci. 1 (4), 300–3007.

Charlton, J., Du, P., Cho, J.-H., Xu, S., 2018. Measuring relative accuracy of malware detectors in the absence of ground truth. In: MILCOM 2018 - 2018 IEEE Military Communications Conference. MILCOM, pp. 450–455. http://dx.doi.org/10.1109/MILCOM.2018.8599730.

Chatzoglou, E., Kambourakis, G., Kolias, C., 2021. Empirical evaluation of attacks against IEEE 802.11 enterprise networks: The AWID3 dataset. IEEE Access 9, 34188–34205. http://dx.doi.org/10.1109/ACCESS.2021.3061609, Dataset available at: https://icsdweb.aegean.gr/awid/.

Chou, D., Jiang, M., 2020. Data-driven network intrusion detection: A taxonomy of challenges and methods. http://dx.doi.org/10.48550/arXiv.2009.07352, arXiv cs.CR.

Cieslak, D.A., Chawla, N.V., Striegel, A., 2006. Combating imbalance in network intrusion datasets.. In: 2006 IEEE International Conference on Granular Computing. pp. 732–737. http://dx.doi.org/10.1109/GRC.2006.1635905.

Claise, B., 2004. Cisco systems NetFlow services export version 9. http://dx.doi.org/10.17487/RFC3954, RFC 3954.

Conti, M., Donadel, D., Turrin, F., 2021. A survey on industrial control system testbeds and datasets for security research. IEEE Commun. Surv. Tutor. 23 (4), 2248–2294. http://dx.doi.org/10.1109/COMST.2021.3094360.

Cordero, C.G., Vasilomanolakis, E., Wainakh, A., Mühlhäuser, M., Nadjm-Tehrani, S., 2021. On generating network traffic datasets with synthetic attacks for intrusion detection. ACM Trans. Priv. Secur. 24 (2), http://dx.doi.org/10.1145/3424155.

Coull, S.E., Monrose, F., Reiter, M.K., Bailey, M., 2009. The challenges of effectively anonymizing network data. In: 2009 Cybersecurity Applications & Technology Conference for Homeland Security. pp. 230–236. http://dx.doi.org/10.1109/CATCH.2009.27.

Creech, G., Hu, J., 2013. Generation of a new IDS test dataset: Time to retire the kdd collection. In: 2013 IEEE Wireless Communications and Networking Conference. WCNC, pp. 4487–4492. http://dx.doi.org/10.1109/WCNC.2013.6555301, Data available at: https://research.unsw.edu.au/projects/adfa-ids-datasets.

Crow, B., Widjaja, I., Kim, J., Sakai, P., 1997. IEEE 802.11 Wireless Local Area networks. IEEE Commun. Mag. 35 (9), 116–126. http://dx.doi.org/10.1109/35.620533.

CyberVAN, 2021. CyberForce scenario data collection. Dataset available at: https://cybervan.peratonlabs.com:9000/cyberforce-scenario-data-sets.

Dadkhah, S., Mahdikhani, H., Danso, P.K., Zohourian, A., Truong, K.A., Ghorbani, A.A., 2022. Towards the development of a realistic multidimensional IoT profiling dataset. In: 2022 19th Annual International Conference on Privacy, Security & Trust. PST, pp. 1–11. http://dx.doi.org/10.1109/PST55820.2022.9851966, Dataset available at: https://www.unb.ca/cic/datasets/iotdataset-2022.html.

Damasevicius, R., Venckauskas, A., Grigaliunas, S., Toldinas, J., Morkevicius, N., Aleliunas, T., Smuikys, P., 2020. LITNET-2020: An annotated real-world network flow dataset for network intrusion detection. Electronics 9 (5), http://dx.doi.org/10.3390/electronics9050800, Dataset available at: https://github.com/Grigaliunas/electronics9050800/.

Das, T., Hamdan, O.A., Shukla, R.M., Sengupta, S., Arslan, E., 2023. UNR-IDD: Intrusion detection dataset using network port statistics. In: 2023 IEEE 20th Consumer Communications & Networking Conference. CCNC, pp. 497–500. http://dx.doi.org/10.1109/CCNC51644.2023.10059640, Dataset available at: https://www.tapadhirdas.com/unr-idd-dataset.

De Keersmaeker, F., Cao, Y., Ndonda, G.K., Sadre, R., 2023. A survey of public IoT datasets for network security research. IEEE Commun. Surv. Tutor. 25 (3), 1808–1840. http://dx.doi.org/10.1109/COMST.2023.3288942.

Değirmenci, E., Sabri Kırca, Y., Özçelik, İ., Yazıcı, A., 2023. ROSIDS23: Network intrusion detection dataset for robot operating system. Data Brief 51, 109739. http://dx.doi.org/10.1016/j.dib.2023.109739, Dataset available at: https://zenodo.org/records/10014434.

Dehlaghi-Ghadim, A., Moghadam, M.H., Balador, A., Hansson, H., 2023. Anomaly detection dataset for industrial control systems. IEEE Access 11, 107982–107996. http://dx.doi.org/10.1109/ACCESS.2023.3320928, Dataset available at: https://www.kaggle.com/datasets/alirezadehlaghi/icssim.

Derbyshire, R., Green, B., Prince, D., Mauthe, A., Hutchison, D., 2018. An analysis of cyber security attack taxonomies. In: 2018 IEEE European Symposium on Security and Privacy Workshops. EuroS&PW, pp. 153–161. http://dx.doi.org/10.1109/EuroSPW.2018.00028.

Dewdney, A.K., 1989. Computer recreations. Sci. Am. 260 (3), 110–113, Available at: http://www.jstor.org/stable/24987184.

D'hooge, L., Verkerken, M., Volckaert, B., Wauters, T., De Turck, F., 2022. Establishing the contaminating effect of metadata feature inclusion in machine-learned network intrusion detection models. In: Cavallaro, L., Gruss, D., Pellegrino, G., Giacinto, G. (Eds.), Detection of Intrusions and Malware, and Vulnerability Assessment. Springer International Publishing, Cham, pp. 23–41. http://dx.doi.org/10.1007/978-3-031-09484-2_2.

D'hooge, L., Verkerken, M., Wauters, T., De Turck, F., Volckaert, B., 2023. Castles built on sand: Observations from classifying academic cybersecurity datasets with minimalist methods. In: Proceedings of the 8th International Conference on Internet of Things, Big Data and Security - IoTBDS. SciTePress, INSTICC, pp. 61–72. http://dx.doi.org/10.5220/0011853300003482.

Duncan, B., 2024. Malware traffic analysis – A site for sharing packet capture (pcap) files and malware samples.. Online. (Accessed 05 December 2024). Available: https://malware-traffic-analysis.net/.

Dundar, M., Krishnapuram, B., Bi, J., Rao, R.B., 2007. Learning classifiers when the training data is not IID. In: Proceedings of the 20th International Joint Conference on Artifical Intelligence. IJCAI '07, Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, pp. 756–761.

Elsayed, M.S., Le-Khac, N.-A., Jurcut, A.D., 2020. InSDN: A novel SDN intrusion dataset. IEEE Access 8, 165263–165284. http://dx.doi.org/10.1109/ACCESS.2020.3022633, Dataset available at: https://aseados.ucd.ie/datasets/SDN/.

Engelen, G., Flood, R., Liu-Thorrold, L., Rimmer, V., Clausen, H., Aspinall, D., Joosen, W., 2022. Poster: Pillars of sand: The current state of datasets in the field of network intrusion detection. In: European Symposium on Security and Privacy. EuroS&P, Vol. 7068716, pp. 1–3. http://dx.doi.org/10.5281/zenodo.

Engelen, G., Rimmer, V., Joosen, W., 2021. Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In: 2021 IEEE Security and Privacy Workshops. SPW, pp. 7–12. http://dx.doi.org/10.1109/SPW53761.2021.00009.

Erhan, D., Anarım, E., 2020. Boğaziçi university distributed denial of service dataset. Data Brief 32, http://dx.doi.org/10.1016/j.dib.2020.106187, Dataset available at: https://data.mendeley.com/datasets/mfnn9bh42m/1.

eSentire, 2023. Cybercrime To Cost The World $9.5 Trillion USD Annually In 2024. Tech. report, (Accessed 13 October 2024). Available at: https://www.esentire.com/web-native-pages/cybercrime-to-cost-the-world-9-5-trillion-usd-annually-in-2024.

Ferrag, M.A., Friha, O., Hamouda, D., Maglaras, L., Janicke, H., 2022. Edge-IIoTset: A new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access 10, 40281–40306. http://dx.doi.org/10.1109/ACCESS.2022.3165809, Dataset available at: https://www.kaggle.com/datasets/mohamedamineferrag/edgeiiotset-cyber-security-dataset-of-iot-iiot.

Ferrag, M.A., Maglaras, L., Moschoyiannis, S., Janicke, H., 2020. Deep learning for cyber security intrusion detection: Approaches, datasets, and comparative study. J. Inf. Secur. Appl. 50, 102419. http://dx.doi.org/10.1016/j.jisa.2019.102419.

Ferriyan, A., Thamrin, A.H., Takeda, K., Murai, J., 2021. Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic. Appl. Sci. 11 (17), Dataset available at: https://doi.org/10.5281/zenodo.6463389.

Finsterbusch, M., Richter, C., Rocha, E., Muller, J.-A., Hanssgen, K., 2014. A survey of payload-based traffic classification approaches. IEEE Commun. Surv. Tutor. 16 (2), 1135–1156. http://dx.doi.org/10.1109/SURV.2013.100613.00161.

Flood, R., Engelen, G., Aspinall, D., Desmet, L., 2024. Bad design smells in benchmark NIDS datasets. In: 2024 IEEE 9th European Symposium on Security and Privacy. EuroS&P, pp. 658–675. http://dx.doi.org/10.1109/EuroSP60621.2024.00042.

Fontugne, R., Borgnat, P., Abry, P., Fukuda, K., 2010. MAWILab: Combining diverse anomaly detectors for automated anomaly labeling and performance benchmarking. In: Proceedings of the 6th International COnference. In: Co-NEXT '10, Association for Computing Machinery, pp. 1–12. http://dx.doi.org/10.1145/1921168.1921179, Dataset available at: http://www.fukuda-lab.org/mawilab/index.html.

Fu, C., Li, Q., Shen, M., Xu, K., 2021. Realtime robust malicious traffic detection via frequency domain analysis. In: Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security. CCS '21, Association for Computing Machinery, New York, NY, USA, pp. 3431–3446. http://dx.doi.org/10.1145/3460120.3484585.

García, S., Grill, M., Stiborek, J., Zunino, A., 2014. An empirical comparison of botnet detection methods. Comput. Secur. 45, 100–123. http://dx.doi.org/10.1016/j.cose.2014.05.011, Dataset available at: https://www.stratosphereips.org/datasets-ctu13.

García, S., Luengo, J., Herrera, F., 2014. Data Preprocessing in Data Mining, first ed. Intelligent Systems Reference Library, Springer, http://dx.doi.org/10.1007/978-3-319-10247-4.

Garcia, S., Parmisano, A., Erquiaga, M.J., 2020. IoT-23: A labeled dataset with malicious and benign IoT network traffic (version 1.0.0) [data set]. http://dx.doi.org/10.5281/zenodo.4743746, Dataset available at: https://www.stratosphereips.org/datasets-iot23.

Geirhos, R., Jacobsen, J.-H., Michaelis, C., Zemel, R., Brendel, W., Bethge, M., Wichmann, F.A., 2020. Shortcut learning in deep neural networks. Nat. Mach. Intell. 2 (11), 665–673. http://dx.doi.org/10.1038/s42256-020-00257-z.

Gharib, A., Sharafaldin, I., Lashkari, A.H., Ghorbani, A.A., 2016. An evaluation framework for intrusion detection dataset. In: 2016 International Conference on Information Science and Security. ICISS, pp. 1–6. http://dx.doi.org/10.1109/ICISSEC.2016.7885840.

Giner-Miguelez, J., Gómez, A., Cabot, J., 2024. Using large language models to enrich the documentation of datasets for machine learning. http://dx.doi.org/10.48550/arXiv.2404.15320.

Goldschmidt, P., Kučera, J., 2024. Windower: Feature extraction for real-time DDoS detection using machine learning. In: NOMS 2024-2024 IEEE Network Operations and Management Symposium. pp. 1–10. http://dx.doi.org/10.1109/NOMS59830.2024.10575699.

Guerra, J.L., Catania, C., Veas, E., 2022. Datasets are not enough: Challenges in labeling network traffic. Comput. Secur. 120, 102810. http://dx.doi.org/10.1016/j.cose.2022.102810.

Guerra-Manzanares, A., Medina-Galindo, J., Bahsi, H., Nõmm, S., 2020. MedBIoT: Generation of an IoT botnet dataset in a medium-sized IoT network. In: Proceedings of the 6th International Conference on Information Systems Security and Privacy - ICISSP. SciTePress, INSTICC, pp. 207–218. http://dx.doi.org/10.5220/0009187802070218, Dataset available at: https://cs.taltech.ee/research/data/medbiot/.

Gümüşbaş, D., Yıldırım, T., Genovese, A., Scotti, F., 2021. A comprehensive survey of databases and deep learning methods for cybersecurity and intrusion detection systems. IEEE Syst. J. 15 (2), 1717–1731. http://dx.doi.org/10.1109/JSYST.2020.2992966.

Haffey, M., Arlitt, M., Williamson, C., 2018. Modeling, analysis, and characterization of periodic traffic on a campus edge network. In: 2018 IEEE 26th International Symposium on Modeling, Analysis, and Simulation of Computer and Telecommunication Systems. MASCOTS, pp. 170–182. http://dx.doi.org/10.1109/MASCOTS.2018.00025.

Haider, W., Hu, J., Slay, J., Turnbull, B., Xie, Y., 2017. Generating realistic intrusion detection system dataset based on fuzzy qualitative modeling. J. Netw. Comput. Appl. 87, 185–192, Dataset available at: https://doi.org/10.26190/unsworks/25289.

Haines, J., Rossey, L., Lippmann, R., Cunningham, R., 2001. Extending the DARPA off-line intrusion detection evaluations. In: Proceedings DARPA Information Survivability Conference and Exposition II. DISCEX'01. Vol. 1, pp. 35–45 vol.1. http://dx.doi.org/10.1109/DISCEX.2001.932190, Dataset available at: https://archive.ll.mit.edu/ideval/data/2000data.html.

He, K., Kim, D.D., Asghar, M.R., 2023. Adversarial machine learning for network intrusion detection systems: A comprehensive survey. IEEE Commun. Surv. Tutor. 25 (1), 538–566. http://dx.doi.org/10.1109/COMST.2022.3233793.

Herzalla, D., Lunardi, W.T., Andreoni, M., 2023. TII-SSRC-23 dataset: Typological exploration of diverse traffic patterns for intrusion detection. IEEE Access 11, 118577–118594. http://dx.doi.org/10.1109/ACCESS.2023.3319213, Dataset available at: https://www.kaggle.com/datasets/daniaherzalla/tii-ssrc-23.

Hindy, H., Brosset, D., Bayne, E., Seeam, A.K., Tachtatzis, C., Atkinson, R., Bellekens, X., 2020. A taxonomy of network threats and the effect of current datasets on intrusion detection systems. IEEE Access 8, 104650–104675. http://dx.doi.org/10.1109/ACCESS.2020.3000179.

Hofstede, R., Hendriks, L., Sperotto, A., Pras, A., 2014. SSH compromise detection using NetFlow/IPFIX. SIGCOMM Comput. Commun. Rev. 44 (5), 20–26. http://dx.doi.org/10.1145/2677046.2677050, Dataset available at: https://www.simpleweb.org/wiki/index.php/SSH_datasets.html.

HogZilla, 2015. The hogzilla dataset. Dataset available at: https://ids-hogzilla.org/dataset/.

Homoliak, I., Malinka, K., Hanacek, P., 2020. ASNM datasets: A collection of network attacks for testing of adversarial classifiers and intrusion detectors. IEEE Access 8, 112427–112453. http://dx.doi.org/10.1109/ACCESS.2020.3001768, Datasets available at: https://www.fit.vutbr.cz/~ihomoliak/asnm/index.html.

Hu, Z., Wang, L., Qi, L., Li, Y., Yang, W., 2020. A novel wireless network intrusion detection method based on adaptive synthetic sampling and an improved convolutional neural network. IEEE Access 8, 195741–195751. http://dx.doi.org/10.1109/ACCESS.2020.3034015.

Hutchins, E.M., Cloppert, M.J., Amin, R.M., et al., 2011. Intelligence-driven computer network defense informed by analysis of adversary campaigns and intrusion kill chains. Lead. Issues Inf. Warf. Secur. Res. 1 (1), 80.

IEEE LAN/MAN Standards Committee, 2020. IEEE standard for low-rate wireless networks. pp. 1–800. http://dx.doi.org/10.1109/IEEESTD.2020.9144691, IEEE Std 802.15.4-2020 (Revision of IEEE Std 802.15. 4- 2015).

Inacio, C.M., Trammell, B., 2010. YAF: yet another flowmeter. In: Proceedings of the 24th International Conference on Large Installation System Administration. LISA '10, USENIX Association, USA, pp. 1–16.

Jacobs, A.S., Beltiukov, R., Willinger, W., Ferreira, R.A., Gupta, A., Granville, L.Z., 2022. AI/ML for network security: The emperor has no clothes. In: Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security. CCS '22, Association for Computing Machinery, New York, NY, USA, pp. 1537–1551. http://dx.doi.org/10.1145/3548606.3560609.

Jazi, H.H., Gonzalez, H., Stakhanova, N., Ghorbani, A.A., 2017. Detecting HTTP-based application layer DoS attacks on web servers in the presence of sampling. Comput. Netw. 121, 25–36. http://dx.doi.org/10.1016/j.comnet.2017.03.018, Dataset available at: https://www.unb.ca/cic/datasets/dos-dataset.html.

Kang, H., Ahn, D.H., Lee, G.M., Yoo, J.D., Park, K.H., Kim, H.K., 2019. IoT network intrusion dataset. http://dx.doi.org/10.21227/q70p-q449, Dataset available at: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset.

Kaur, B., Dadkhah, S., Shoeleh, F., Neto, E.C.P., Xiong, P., Iqbal, S., Lamontagne, P., Ray, S., Ghorbani, A.A., 2023. Internet of things (IoT) security dataset evolution: Challenges and future directions. Internet Things 22, 100780. http://dx.doi.org/10.1016/j.iot.2023.100780.

Kent, A.D., 2016. Cyber security data sources for dynamic network research. In: Dynamic Networks and Cyber-Security. World Scientific, pp. 37–65. http://dx.doi.org/10.1142/9781786340757_0002, Dataset available at: https://csr.lanl.gov/data/cyber1/.

Kenyon, A., Deka, L., Elizondo, D., 2020. Are public intrusion datasets fit for purpose characterising the state of the art in intrusion event datasets. Comput. Secur. 99, 102022. http://dx.doi.org/10.1016/j.cose.2020.102022.

Kitchenham, B., Charters, S., 2007. Guidelines for Performing Systematic Literature Reviews in Software Engineering. Technical Report, Software Engineering Group, School of Computer Science and Mathematics, Keele University, ST5 5BG, Keele, UK.

Koay, A.M.Y., Ko, R.K.L., Hettema, H., Radke, K., 2023. Machine learning in industrial control system (ICS) security: current landscape, opportunities and challenges. J. Intell. Inf. Syst. 60 (2), 377–405. http://dx.doi.org/10.1007/s10844-022-00753-1.

Kolias, C., Kambourakis, G., Stavrou, A., Gritzalis, S., 2016. Intrusion detection in 802.11 networks: Empirical evaluation of threats and a public dataset. IEEE Commun. Surv. Tutor. 18 (1), 184–208. http://dx.doi.org/10.1109/COMST.2015.2402161, Dataset available at: https://icsdweb.aegean.gr/awid/.

Komisarek, M., Pawlicki, M., Mihailescu, M.-E., Mihai, D., Carabas, M., Kozik, R., Choraś, M., 2022. A novel, refined dataset for real-time network intrusion detection. In: Proceedings of the 17th International Conference on Availability, Reliability and Security. ARES '22, Association for Computing Machinery, pp. 1–8. http://dx.doi.org/10.1145/3538969.3544486, Dataset available at: https://www.kaggle.com/datasets/h2020simargl/simargl2022.

Komisarek, M., Pawlicki, M., Simic, T., Kavcnik, D., Kozik, R., Choraś, M., 2023. Modern NetFlow network dataset with labeled attacks and detection methods. In: Proceedings of the 18th International Conference on Availability, Reliability and Security. ARES '23, Association for Computing Machinery, New York, NY, USA, pp. 1–8. http://dx.doi.org/10.1145/3600160.3605094, Dataset available at: https://www.kaggle.com/datasets/ittibydgoszcz/appraise-h2020-real-labelled-netflow-dataset.

Koroniotis, N., Moustafa, N., Sitnikova, E., Turnbull, B., 2019. Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: Bot-IoT dataset. Future Gener. Comput. Syst. 100, 779–796. http://dx.doi.org/10.1016/j.future.2019.05.041, Dataset available at: https://research.unsw.edu.au/projects/bot-iot-dataset.

Kotenko, I., Gaifulina, D., Zelichenok, I., 2022. Systematic literature review of security event correlation methods. IEEE Access 10, 43387–43420. http://dx.doi.org/10.1109/ACCESS.2022.3168976.

Kumar, P., Liu, J., Md Tayeen, A.S., Misra, S., Cao, H., Harikumar, J., Perez, O., 2023. FLNET2023: Realistic network intrusion detection dataset for federated learning. In: MILCOM 2023 - 2023 IEEE Military Communications Conference. MILCOM, pp. 345–350. http://dx.doi.org/10.1109/MILCOM58377.2023.10356272, Dataset available at: https://github.com/nsol-nmsu/FML-Network.

Kumar, V., Sinha, D., 2023. Synthetic attack data generation model applying generative adversarial network for intrusion detection. Comput. Secur. 125, 103054. http://dx.doi.org/10.1016/j.cose.2022.103054.

Landauer, M., Frank, M., Skopik, F., Hotwagner, W., Wurzenberger, M., Rauber, A., 2021. Kyoushi log data set. http://dx.doi.org/10.5281/zenodo.5779411.

Landauer, M., Skopik, F., Frank, M., Hotwagner, W., Wurzenberger, M., Rauber, A., 2023. Maintainable log datasets for evaluation of intrusion detection systems. IEEE Trans. Dependable Secur. Comput. 20 (4), 3466–3482. http://dx.doi.org/10.1109/TDSC.2022.3201582, Dataset available at: https://zenodo.org/records/5789064. NetFlow data at: https://zenodo.org/records/13168643.

Lantz, B., Heller, B., McKeown, N., 2010. A network in a laptop: rapid prototyping for software-defined networks. In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. In: Hotnets-IX, Association for Computing Machinery, New York, NY, USA, pp. 1–6. http://dx.doi.org/10.1145/1868447.1868466.

Lanvin, M., Gimenez, P.-F., Han, Y., Majorczyk, F., Mé, L., Totel, É., 2023. Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes. In: Kallel, S., Jmaiel, M., Zulkernine, M., Hadj Kacem, A., Cuppens, F., Cuppens, N. (Eds.), Risks and Security of Internet and Systems. Springer Nature Switzerland, Cham, pp. 18–33. http://dx.doi.org/10.1007/978-3-031-31108-6_2.

Lashkari, A.H., 2016. CICFlowMeter. Canadian Institute for Cybersecurity, (Accessed 12 March 2024). Available at: https://www.unb.ca/cic/research/applications.html#CICFlowMeter.

Lashkari, A.H., 2021. CICFlowMeter extracted features and descriptions. [Online]. (Accessed 16 April 2024). Available at: https://github.com/ahlashkari/CICFlowMeter/blob/master/ReadMe.txt.

Law, E., von Ahn, L., 2011. Human Computation, first ed. Morgan & Claypool Publishers, http://dx.doi.org/10.2200/S00371ED1V01Y201107AIM013.

Lawrence, H., Ezeobi, U., Tauil, O., Nosal, J., Redwood, O., Zhuang, Y., Bloom, G., 2022. CUPID: A labeled dataset with pentesting for evaluation of network intrusion detection. J. Syst. Archit. 129, 102621. http://dx.doi.org/10.1016/j.sysarc.2022.102621, Dataset available at: https://www.cupid.directory.

Layeghy, S., Gallagher, M., Portmann, M., 2024. Benchmarking the benchmark — Comparing synthetic and real-world Network IDS datasets. J. Inf. Secur. Appl. 80, 103689. http://dx.doi.org/10.1016/j.jisa.2023.103689.

Lewandowski, B., 2023. Guidelines and a framework to improve the delivery of network intrusion detection datasets. In: Proceedings of the 20th International Conference on Security and Cryptography - Volume 1: SECRYPT. SciTePress, pp. 649–658. http://dx.doi.org/10.5220/0012052300003555.

Lippmann, R., Fried, D., Graf, I., Haines, J., Kendall, K., McClung, D., Weber, D., Webster, S., Wyschogrod, D., Cunningham, R., Zissman, M., 2000a. Evaluating intrusion detection systems: the 1998 DARPA off-line intrusion detection evaluation. In: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00. Vol. 2, pp. 12–26 vol.2. http://dx.doi.org/10.1109/DISCEX.2000.821506, Dataset available at: https://www.ll.mit.edu/r-d/datasets/1998-darpa-intrusion-detection-evaluation-dataset.

Lippmann, R., Haines, J.W., Fried, D.J., Korba, J., Das, K., 2000b. The 1999 DARPA off-line intrusion detection evaluation. Comput. Netw. 34 (4), 579–595. http://dx.doi.org/10.1016/S1389-1286(00)00139-0, Recent Advances in Intrusion Detection Systems. Dataset available at: https://www.ll.mit.edu/r-d/datasets/1999-darpa-intrusion-detection-evaluation-dataset.

Lipton, Z.C., Steinhardt, J., 2019. Research for Practice: Troubling Trends in Machine-Learning Scholarship. Commun. ACM 62 (6), 45–53. http://dx.doi.org/10.1145/3316774.

Liu, L., Engelen, G., Lynar, T., Essam, D., Joosen, W., 2022. Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018. In: 2022 IEEE Conference on Communications and Network Security. CNS, IEEE, pp. 254–262. http://dx.doi.org/10.1109/CNS56114.2022.9947235, Improved CIC-IDS2017 and CSE-CIC-IDS2018 datasets available at: https://intrusion-detection.distrinet-research.be/CNS2022/index.html.

Liu, Z., Thapa, N., Shaver, A., Roy, K., Siddula, M., Yuan, X., Yu, A., 2021b. Using embedded feature selection and CNN for classification on CCD-INID-V1—A new IoT dataset. Sensors 21 (14), http://dx.doi.org/10.3390/s21144834, Data available at: https://github.com/Innovate3000/CCD-INID-V1/.

Liu, L., Wang, P., Lin, J., Liu, L., 2021a. Intrusion detection of imbalanced network traffic based on machine learning and deep learning. IEEE Access 9, 7550–7563. http://dx.doi.org/10.1109/ACCESS.2020.3048198.

Lukaseder, T., 2017. 2017-SUEE-data-set. Dataset available at: https://github.com/vs-uulm/2017-SUEE-data-set.

Maciá-Fernández, G., Camacho, J., Magán-Carrión, R., García-Teodoro, P., Therón, R., 2018. UGR'16: A new dataset for the evaluation of cyclostationarity-based network IDSs. Comput. Secur. 73, 411–424. http://dx.doi.org/10.1016/j.cose.2017.11.004, Dataset available at: https://nesg.ugr.es/nesg-ugr16/.

Mahfouz, A., Abuhussein, A., Venugopal, D., Shiva, S., 2020. Ensemble classifiers for network intrusion detection using a novel network attack dataset. Futur. Internet 12 (11), http://dx.doi.org/10.3390/fi12110180, Dataset available at: https://www.researchgate.net/publication/344478320_GTCS_Network_Attack_Dataset.

Mahoney, M.V., Chan, P.K., 2003. An analysis of the 1999 DARPA/Lincoln laboratory evaluation data for network anomaly detection. In: Recent Advances in Intrusion Detection. Springer Berlin Heidelberg, pp. 220–237. http://dx.doi.org/10.1007/978-3-540-45248-5_13.

Małowidzki, M., Berezinski, P., Mazur, M., 2015. Network intrusion detection: Half a kingdom for a good dataset. In: Proceedings of NATO STO SAS-139 Workshop. Portugal, pp. 1–6.

Mann, H.B., Whitney, D.R., 1947. On a test of whether one of two random variables is stochastically larger than the other. Ann. Math. Stat. 18 (1), 50–60. http://dx.doi.org/10.1214/aoms/1177730491.

Manocchio, L.D., Layeghy, S., Portmann, M., 2021. FlowGAN - Synthetic network flow generation using generative adversarial networks. In: 2021 IEEE 24th International Conference on Computational Science and Engineering. CSE, pp. 168–176. http://dx.doi.org/10.1109/CSE53436.2021.00033.

Masana, M., Liu, X., Twardowski, B., Menta, M., Bagdanov, A.D., van de Weijer, J., 2023. Class-incremental learning: Survey and performance evaluation on image classification. IEEE Trans. Pattern Anal. Mach. Intell. 45 (5), 5513–5533. http://dx.doi.org/10.1109/TPAMI.2022.3213473.

Mauro, M.D., Galatro, G., Fortino, G., Liotta, A., 2021. Supervised feature selection techniques in network intrusion detection: A critical review. Eng. Appl. Artif. Intell. 101, 104216. http://dx.doi.org/10.1016/j.engappai.2021.104216.

MAWI Working Group, 2024. MAWI working group traffic archive. Online. (Accessed 14 June 2024). Available at: http://mawi.wide.ad.jp/mawi/.

McHugh, J., 2000. 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. 3 (4), 262–294. http://dx.doi.org/10.1145/382912.382923.

Meidan, Y., Bohadana, M., Mathov, Y., Mirsky, Y., Shabtai, A., Breitenbacher, D., Elovici, Y., 2018. N-BaIoT—Network-based detection of IoT botnet attacks using deep autoencoders. IEEE Pervasive Comput. 17 (3), 12–22, Dataset available at: https://doi.org/10.24432/C5RC8J.

Meng, W., Li, W., Su, C., Zhou, J., Lu, R., 2018. Enhancing trust management for wireless intrusion detection via traffic sampling in the era of big data. IEEE Access 6, 7234–7243. http://dx.doi.org/10.1109/ACCESS.2017.2772294.

Mills, R., 2022. Luflow network intrusion detection data set. http://dx.doi.org/10.34740/kaggle/dsv/11194052, Dataset available at: https://www.kaggle.com/datasets/mryanm/luflow-network-intrusion-detection-data-set.

Mirsky, Y., Doitshman, T., Elovici, Y., Shabtai, A., 2018. Kitsune: An ensemble of autoencoders for online network intrusion detection. In: The Network and Distributed System Security Symposium (NDSS) 2018. pp. 1–15, Dataset available at: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune.

MITRE, 2024. MITRE ATT&CK®. Online. (Accessed 11 April 2024). Available https://attack.mitre.org/.

MITRE Corporation, 2024. MITRE caldera: A scalable, automated adversary emulation platform. Online. (Accessed 12 July 2024). Available at: https://github.com/mitre/caldera.

Mittal, M., Kumar, K., Behal, S., 2023. Deep learning approaches for detecting DDoS attacks: a systematic review. Soft Comput. 27 (18), 13039–13075. http://dx.doi.org/10.1007/s00500-021-06608-1.

Molina-Coronado, B., Mori, U., Mendiburu, A., Miguel-Alonso, J., 2020. Survey of network intrusion detection methods from the perspective of the knowledge discovery in databases process. IEEE Trans. Netw. Serv. Manag. 17 (4), 2451–2479. http://dx.doi.org/10.1109/TNSM.2020.3016246.

Moustafa, N., 2021. A new distributed architecture for evaluating AI-based security systems at the edge: Network TON_IoT datasets. Sustain. Cities Soc. 72, 102994. http://dx.doi.org/10.1016/j.scs.2021.102994.

Moustafa, N., 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, pp. 1–6. http://dx.doi.org/10.1109/MilCIS.2015.7348942, Dataset available at: https://research.unsw.edu.au/projects/unsw-nb15-dataset.

Mukherjee, B., Heberlein, L., Levitt, K., 1994. Network intrusion detection. IEEE Netw. 8 (3), 26–41. http://dx.doi.org/10.1109/65.283931.

Myneni, S., Chowdhary, A., Sabur, A., Sengupta, S., Agrawal, G., Huang, D., Kang, M., 2020. DAPT 2020 - constructing a benchmark dataset for advanced persistent threats. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (Eds.), Deployable Machine Learning for Security Defense. Springer International Publishing, pp. 138–163. http://dx.doi.org/10.1007/978-3-030-59621-7_8, Dataset available at: https://www.kaggle.com/datasets/sowmyamyneni/dapt2020.

Myneni, S., Jha, K., Sabur, A., Agrawal, G., Deng, Y., Chowdhary, A., Huang, D., 2023. Unraveled — A semi-synthetic dataset for advanced persistent threats. Comput. Netw. 227, 109688. http://dx.doi.org/10.1016/j.comnet.2023.109688, Dataset available at: https://gitlab.com/asu22/unraveled/.

Nadeem, A., Vos, D., Cao, C., Pajola, L., Dieck, S., Baumgartner, R., Verwer, S., 2023. SoK: Explainable machine learning for computer security applications. In: 2023 IEEE 8th European Symposium on Security and Privacy. EuroS&P, pp. 221–240. http://dx.doi.org/10.1109/EuroSP57164.2023.00022.

NetFlow Version 9, 2011. NetFlow Version 9 Flow-Record Format. Tech. rep., https://www.cisco.com/en/US/technologies/tk648/tk362/technologies_white_paper09186a00800a3db9.pdf.

Neto, E.C.P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., Ghorbani, A.A., 2023. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors 23 (13), http://dx.doi.org/10.3390/s23135941, Dataset available at: https://www.unb.ca/cic/datasets/iotdataset-2023.html.

NETRESEC AB, 2024. Capture files from mid-atlantic CCDC. Online. (Accessed 10 June 2024). Available at: https://www.netresec.com/?page=MACCDC.

Ngo, V.-D., Vuong, T.-C., Van Luong, T., Tran, H., 2024. Machine learning-based intrusion detection: feature selection versus feature extraction. Clust. Comput. 27 (3), 2365–2379. http://dx.doi.org/10.1007/s10586-023-04089-5.

Northcutt, C.G., Athalye, A., Mueller, J., 2021. Pervasive label errors in test sets destabilize machine learning benchmarks. http://dx.doi.org/10.48550/arXiv.2103.14749.

Ntop, 2023. nProbe™ – An extensible NetFlow v5/v9/IPFIX probe for IPv4/v6. [Online]. (Accessed 12 March 2024). Available at: https://www.ntop.org/products/netflow/nprobe/.

Nucci, A., Sridharan, A., Taft, N., 2005. The problem of synthetically generating IP traffic matrices: initial recommendations. ACM SIGCOMM Comput. Commun. Rev. 35 (3), 19–32. http://dx.doi.org/10.1145/1070873.1070876.

Open Networking Foundation, 2015. OpenFlow switch specification. https://opennetworking.org/wp-content/uploads/2014/10/openflow-switch-v1.5.1.pdf.

Pang, R., Allman, M., Bennett, M., Lee, J., Paxson, V., Tierney, B., 2005. A first look at modern enterprise traffic. In: Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement. IMC '05, USENIX Association, USA, p. 2, Dataset available at: http://www.icir.org/enterprise-tracing/.

Papaioannou, M., Karageorgou, M., Mantas, G., Sucasas, V., Essop, I., Rodriguez, J., Lymberopoulos, D., 2022. A survey on security threats and countermeasures in internet of medical things (IoMT). Trans. Emerg. Telecommun. Technol. 33 (6), e4049. http://dx.doi.org/10.1002/ett.4049.

Parisi, G.I., Kemker, R., Part, J.L., Kanan, C., Wermter, S., 2019. Continual lifelong learning with neural networks: A review. Neural Netw. 113, 54–71. http://dx.doi.org/10.1016/j.neunet.2019.01.012.

Patel, N.D., Mehtre, B.M., Wankar, R., 2023. Od-ids2022: generating a new offensive defensive intrusion detection dataset for machine learning-based attack classification. Int. J. Inf. Technol. 15 (8), 4349–4363. http://dx.doi.org/10.1007/s41870-023-01464-8, Dataset available only on request..

Paxson, V., 1999. Bro: a system for detecting network intruders in real-time. Comput. Netw. 31 (23–24), 2435–2463, https://zeek.org/.

Pendlebury, F., Pierazzi, F., Jordaney, R., Kinder, J., Cavallaro, L., et al., 2019. TESSERACT: Eliminating experimental bias in malware classification across space and time. In: Proceedings of the 28th USENIX Security Symposium. USENIX Association, pp. 729–746.

Perona, I., Gurrutxaga, I., Arbelaitz, O., Martín, J.I., Muguerza, J., Pérez, J.M., 2008. Service-independent payload analysis to improve intrusion detection in network traffic. In: Proceedings of the 7th Australasian Data Mining Conference - Volume 87. AusDM '08, Australian Computer Society, Inc., pp. 171–178, Dataset available at: http://www.sc.ehu.es/acwaldap/gureKddcup/.

Phaal, P., Lavine, M., 2004. Sflow version 5. [Online]. (Accessed 11 March 2024). Available at: https://sflow.org/sflow_version_5.txt.

Pinto, R., 2020. The OPC UA dataset. http://dx.doi.org/10.21227/ychv-6c68, Dataset available at: https://digi2-feup.github.io/OPCUADataset/.

Qosient, L., 2022. Argus. [Online]. (Accessed 15 April 2024). Available at: https://openargus.org/.

Riera, T.S., Higuera, J.-R.B., Higuera, J.B., Herraiz, J.-J.M., Montalvo, J.-A.S., 2022. A new multi-label dataset for web attacks CAPEC classification using machine learning techniques. Comput. Secur. 120, 102788, Dataset available at: https://doi.org/10.7910/DVN/OGOIXX.

Riley, G.F., Henderson, T.R., 2010. The ns-3 network simulator. In: Modeling and Tools for Network Simulation. Springer Berlin Heidelberg, Berlin, Heidelberg, pp. 15–34. http://dx.doi.org/10.1007/978-3-642-12331-3_2.

Ring, M., Dallmann, A., Landes, D., Hotho, A., 2017a. IP2Vec: Learning similarities between IP addresses. In: 2017 IEEE International Conference on Data Mining Workshops. ICDMW, pp. 657–666. http://dx.doi.org/10.1109/ICDMW.2017.93.

Ring, M., Schlör, D., Landes, D., Hotho, A., 2019a. Flow-based network traffic generation using Generative Adversarial Networks. Comput. Secur. 82, 156–172. http://dx.doi.org/10.1016/j.cose.2018.12.012.

Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A., 2017b. Creation of flow-based data sets for intrusion detection. J. Inf. Warf. 16 (4), 41–54, https://www.jstor.org/stable/26504117. Dataset available at: https://www.hs-coburg.de/cidds.

Ring, M., Wunderlich, S., Grüdl, D., Landes, D., Hotho, A., 2017c. Flow-based benchmark data sets for intrusion detection. In: Proceedings of the 16th European Conference on Cyber Warfare and Security. ECCWS, ACPI, pp. 361–369, Dataset available at: https://www.hs-coburg.de/cidds.

Ring, M., Wunderlich, S., Scheuring, D., Landes, D., Hotho, A., 2019b. A survey of network-based intrusion detection data sets. Comput. Secur. 86, 147–167. http://dx.doi.org/10.1016/j.cose.2019.06.005.

Rosay, A., Carlier, F., Cheval, E., LeRoux, P., 2022. From CIC-IDS2017 to LYCOS-IDS2017: A corrected dataset for better performance. In: IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology. In: WI-IAT '21, Association for Computing Machinery, New York, NY, USA, pp. 570–575. http://dx.doi.org/10.1145/3486622.3493973, Dataset available at: https://lycos-ids.univ-lemans.fr/.

Saad, S., Traore, I., Ghorbani, A., Sayed, B., Zhao, D., Lu, W., Felix, J., Hakimian, P., 2011. Detecting P2P botnets through network behavior analysis and machine learning. In: 2011 Ninth Annual International Conference on Privacy, Security and Trust. pp. 174–180. http://dx.doi.org/10.1109/PST.2011.5971980, Dataset available at: https://onlineacademiccommunity.uvic.ca/isot/2022/11/27/botnet-and-ransomware-detection-datasets/.

Sangster, B., O'Connor, T.J., Cook, T., Fanelli, R., Dean, E., Adams, W.J., Morrell, C., Conti, G., 2009. Toward instrumenting network warfare competitions to generate labeled datasets. In: Proceedings of the 2nd Conference on Cyber Security Experimentation and Test. CSET '09, USENIX Association, USA, p. 9, Dataset available at: https://www.westpoint.edu/centers-and-research/cyber-research-center/data-sets.

Santanna, J., van Rijswijk-Deij, R., Hofstede, R., Sperotto, A., Wierbosch, M., Zambenedetti Granville, L., Pras, A., 2015. Booters - an analysis of ddos-as-a-service attacks. In: IFIP/IEEE International Symposium on Integrated Network Management. IM, pp. 243–251. http://dx.doi.org/10.1109/INM.2015.7140298, Dataset available at: https://www.simpleweb.org/wiki/index.php/Traces.html#Booters_-_An_analysis_of_DDoS-as-a-Service_Attacks.

Santos, P.R.d.P.F., Resende, P.A.A., Gondim, J.J.C., Drummond, A.C., 2025. Towards robust cyber attack taxonomies: A survey with requirements, structures, and assessment. ACM Comput. Surv. 57 (8), http://dx.doi.org/10.1145/3717606.

Sarhan, M., Layeghy, S., Moustafa, N., Portmann, M., 2021. NetFlow datasets for machine learning-based network intrusion detection systems. In: Deze, Z., Huang, H., Hou, R., Rho, S., Chilamkurti, N. (Eds.), Big Data Technologies and Applications. Springer International Publishing, Cham, pp. 117–135. http://dx.doi.org/10.1007/978-3-030-72802-1_9, Datasets available at: https://staff.itee.uq.edu.au/marius/NIDS_datasets/.

Sarhan, M., Layeghy, S., Portmann, M., 2022a. Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection. Big Data Res. 30, 100359. http://dx.doi.org/10.1016/j.bdr.2022.100359.

Sarhan, M., Layeghy, S., Portmann, M., 2022b. Towards a standard feature set for network intrusion detection system datasets. Mob. Netw. Appl. 27 (1), 357–370. http://dx.doi.org/10.1007/s11036-021-01843-0, Datasets available at: https://staff.itee.uq.edu.au/marius/NIDS_datasets/.

Sarica, A.K., Angin, P., 2020. A novel SDN dataset for intrusion detection in IoT networks. In: 2020 16th International Conference on Network and Service Management. CNSM, pp. 1–5. http://dx.doi.org/10.23919/CNSM50824.2020.9269042, Dataset available at: https://github.com/AlperKaan35/SDN-Dataset.

Schoen, A., Blanc, G., Gimenez, P.-F., Han, Y., Majorczyk, F., Me, L., 2024. A tale of two methods: Unveiling the limitations of GAN and the rise of Bayesian networks for synthetic network traffic generation. In: 2024 IEEE European Symposium on Security and Privacy Workshops. EuroS&PW, pp. 273–286. http://dx.doi.org/10.1109/EuroSPW61312.2024.00036.

Serinelli, B.M., 2023. Toward generating a DoS and scan statistical network traffic metrics for building intrusion detection solution based on machine and deep learning: I-Sec-IDS datasets. AIP Conf. Proc. 2724 (1), 040010. http://dx.doi.org/10.1063/5.0141860, Dataset available at: https://gitlab.unige.ch/Benedetto.Serinelli/i-sec-dataset/.

Sethuraman, S.C., Jadapalli, T.G., Sudhakaran, D.P.V., Mohanty, S.P., 2023. Flow based containerized honeypot approach for network traffic analysis: An empirical study. Comput. Sci. Rev. 50, 100600. http://dx.doi.org/10.1016/j.cosrev.2023.100600.

Settles, B., 2009. Active Learning Literature Survey. Technical Report 1648, Department of Computer Sciences, University of Wisconsin-Madison, Madison, Wisconsin, US, Available at: http://digital.library.wisc.edu/1793/60660.

Sharafaldin, I., Gharib, A., Lashkari, A.H., Ghorbani, A.A., et al., 2018a. Towards a reliable intrusion detection benchmark dataset. Softw. Netw. 2018 (1), 177–200.

Sharafaldin, I., Habibi Lashkari, A., Ghorbani, A., 2018b. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In: 4th International Conference on Information Systems Security and Privacy. pp. 108–116. http://dx.doi.org/10.5220/0006639801080116, CIC-IDS2017 dataset available at: https://www.unb.ca/cic/datasets/ids-2017.html, CSE-CIC2018 at: https://www.unb.ca/cic/datasets/ids-2018.html.

Sharafaldin, I., Lashkari, A.H., Hakak, S., Ghorbani, A.A., 2019. Developing realistic distributed denial of service (ddos) attack dataset and taxonomy. In: 2019 International Carnahan Conference on Security Technology. ICCST, pp. 1–8. http://dx.doi.org/10.1109/CCST.2019.8888419, Dataset available at: https://www.unb.ca/cic/datasets/ddos-2019.html.

Shiravi, A., Shiravi, H., Tavallaee, M., Ghorbani, A.A., 2012. Toward developing a systematic approach to generate benchmark datasets for intrusion detection. Comput. Secur. 31 (3), 357–374. http://dx.doi.org/10.1016/j.cose.2011.12.012, Dataset available at: https://www.unb.ca/cic/datasets/ids.html.

Shorten, C., Khoshgoftaar, T.M., 2019. A survey on image data augmentation for deep learning. J. Big Data 6 (1), 60. http://dx.doi.org/10.1186/s40537-019-0197-0.

Silva, J.M.C., Carvalho, P., Lima, S.R., 2017. Inside packet sampling techniques: exploring modularity to enhance network measurements. Int. J. Commun. Syst. 30 (6), e3135. http://dx.doi.org/10.1002/dac.3135.

Silva, J.V.V., de Oliveira, N.R., Medeiros, D.S.V., Lopez, M.A., Mattos, D.M.F., 2022. A statistical analysis of intrinsic bias of network security datasets for training machine learning mechanisms. Ann. Telecommun. 77 (7), 555–571. http://dx.doi.org/10.1007/s12243-021-00904-5.

Silva, J.V.V., Lopez, M.A., Mattos, D.M.F., 2020. Attackers are not stealthy: Statistical analysis of the well-known and infamous KDD network security dataset. In: 2020 4th Conference on Cloud and Internet of Things. CIoT, pp. 1–8. http://dx.doi.org/10.1109/CIoT50422.2020.9244289.

Sommer, R., Paxson, V., 2010. Outside the closed world: On using machine learning for network intrusion detection. In: 2010 IEEE Symposium on Security and Privacy. pp. 305–316. http://dx.doi.org/10.1109/SP.2010.25.

Sommers, J., Yegneswaran, V., Barford, P., 2004. A framework for malicious workload generation. In: Proceedings of the 4th ACM SIGCOMM Conference on Internet Measurement. IMC '04, Association for Computing Machinery, New York, NY, USA, pp. 82–87. http://dx.doi.org/10.1145/1028788.1028799.

Song, J., Takakura, H., Okabe, Y., Eto, M., Inoue, D., Nakao, K., 2011. Statistical analysis of honeypot data and building of kyoto 2006+ dataset for NIDS evaluation. In: Proceedings of the First Workshop on Building Analysis Datasets and Gathering Experience Returns for Security. BADGERS '11, Association for Computing Machinery, New York, NY, USA, pp. 29–36. http://dx.doi.org/10.1145/1978672.1978676, Dataset available at: https://www.takakura.com/Kyoto_data/.

Soukup, D., Tisovčík, P., Hynek, K., Čejka, T., 2021. Towards evaluating quality of datasets for network traffic domain. In: 2021 17th International Conference on Network and Service Management. CNSM, pp. 264–268. http://dx.doi.org/10.23919/CNSM52442.2021.9615601.

Sperotto, A., Sadre, R., van Vliet, F., Pras, A., 2009. A labeled data set for flow-based intrusion detection. In: IP Operations and Management. Springer Berlin Heidelberg, pp. 39–50. http://dx.doi.org/10.1007/978-3-642-04968-2_4, Dataset available at: https://www.simpleweb.org/wiki/index.php/Traces.html#Labeled_Dataset_for_Intrusion_Detection.

Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B., 2010. An overview of IP flow-based intrusion detection. IEEE Commun. Surv. Tutor. 12 (3), 343–356. http://dx.doi.org/10.1109/SURV.2010.032210.00054.

Stolfo, S., Fan, W., Lee, W., Prodromidis, A., Chan, P., 2000. Cost-based modeling for fraud and intrusion detection: results from the JAM project. In: Proceedings DARPA Information Survivability Conference and Exposition. DISCEX'00. Vol. 2, pp. 130–144 vol.2. http://dx.doi.org/10.1109/DISCEX.2000.821515, Dataset available at: http://kdd.ics.uci.edu/databases/kddcup99/kddcup99.html.

Strandberg, P.E., Söderman, D., Dehlaghi-Ghadim, A., Leon, M., Markovic, T., Punnekkat, S., Moghadam, M.H., Buffoni, D., 2023. The westermo network traffic data set. Data Brief 50, 109512. http://dx.doi.org/10.1016/j.dib.2023.109512, Dataset available at: https://github.com/westermo/network-traffic-dataset.

Stratosphere Research Laboratory, 2024. Stratosphere research laboratory. [Online]. (Accessed 19 April 2024). Available at: https://www.stratosphereips.org.

Student, 1908. The probable error of a mean. Biometrika 6, 1–25.

Szumelda, P., Orzechowski, N., Rawski, M., Janicki, A., 2022. VHS-22 – A very heterogeneous set of network traffic data for threat detection. In: Proceedings of the 2022 European Interdisciplinary Cybersecurity Conference. EICC '22, Association for Computing Machinery, New York, NY, USA, pp. 72–78. http://dx.doi.org/10.1145/3528580.3532843, Dataset available at: https://www.kaggle.com/datasets/h2020simargl/vhs-22-network-traffic-dataset.

Tavallaee, M., Bagheri, E., Lu, W., Ghorbani, A.A., 2009. 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. http://dx.doi.org/10.1109/CISDA.2009.5356528, Dataset available at: https://www.unb.ca/cic/datasets/nsl.html.

Tcpreplay, 2024. Tcpreplay - pcap editing and replaying utilities. Online. (Accessed 14 August 2024). Available at: https://tcpreplay.appneta.com/.

Teixeira, M.A., Salman, T., Zolanvari, M., Jain, R., Meskin, N., Samaka, M., 2018. SCADA system testbed for cybersecurity research using machine learning approach. Futur. Internet 10 (8), http://dx.doi.org/10.3390/fi10080076, Dataset available at: https://www.cse.wustl.edu/~jain/iiot/index.html.

Thabtah, F., Hammoud, S., Kamalov, F., Gonsalves, A., 2020. Data imbalance in classification: Experimental evaluation. Inform. Sci. 513, 429–441. http://dx.doi.org/10.1016/j.ins.2019.11.004.

Thakkar, A., Lohiya, R., 2022. A survey on intrusion detection system: feature selection, model, performance measures, application perspective, challenges, and future research directions. Artif. Intell. Rev. 55 (1), 453–563. http://dx.doi.org/10.1007/s10462-021-10037-9.

Tobi, A.M.A., Duncan, I., 2018. KDD 1999 generation faults: a review and analysis. J. Cyber Secur. Technol. 2 (3–4), 164–200. http://dx.doi.org/10.1080/23742917.2018.1518061.

Turcotte, M.J., Kent, A.D., Hash, C., 2019. Unified host and network data set. In: Data Science for Cyber-Security. World Scientific, pp. 1–22. http://dx.doi.org/10.1142/9781786345646_001, Dataset available at: https://csr.lanl.gov/data/2017/.

Turukmane, A.V., Devendiran, R., 2024. M-MultiSVM: An efficient feature selection assisted network intrusion detection system using machine learning. Comput. Secur. 137, 103587. http://dx.doi.org/10.1016/j.cose.2023.103587.

Ullah, I., Mahmoud, Q.H., 2020. A scheme for generating a dataset for anomalous activity detection in IoT networks. In: Goutte, C., Zhu, X. (Eds.), Advances in Artificial Intelligence. Springer International Publishing, Cham, pp. 508–520. http://dx.doi.org/10.1007/978-3-030-47358-7_52, Dataset available at: https://sites.google.com/view/iot-network-intrusion-dataset/.

Umer, M.F., Sher, M., Bi, Y., 2017. Flow-based intrusion detection: Techniques and challenges. Comput. Secur. 70, 238–254. http://dx.doi.org/10.1016/j.cose.2017.05.009.

Vasilomanolakis, E., Cordero, C.G., Milanov, N., Mühlhäuser, M., 2016. Towards the creation of synthetic, yet realistic, intrusion detection datasets. In: NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium. pp. 1209–1214. http://dx.doi.org/10.1109/NOMS.2016.7502989.

Viegas, E.K., Santin, A.O., Oliveira, L.S., 2017. Toward a reliable anomaly-based intrusion detection in real-world environments. Comput. Netw. 127, 200–216. http://dx.doi.org/10.1016/j.comnet.2017.08.013, Dataset available at: https://secplab.ppgia.pucpr.br/?q=trabid.

Vinayakumar, R., Alazab, M., Soman, K.P., Poornachandran, P., Al-Nemrat, A., Venkatraman, S., 2019. Deep learning approach for intelligent intrusion detection system. IEEE Access 7, 41525–41550. http://dx.doi.org/10.1109/ACCESS.2019.2895334.

Vormayr, G., Fabini, J., Zseby, T., 2020. Why are my flows different? A tutorial on flow exporters. IEEE Commun. Surv. Tutor. 22 (3), 2064–2103. http://dx.doi.org/10.1109/COMST.2020.2989695.

Wang, X., 2022. ENIDrift: A fast and adaptive ensemble system for network intrusion detection under real-world drift. In: Proceedings of the 38th Annual Computer Security Applications Conference. ACSAC '22, Association for Computing Machinery, New York, NY, USA, pp. 785–798. http://dx.doi.org/10.1145/3564625.3567992, GitHub repository available at: https://github.com/X1anWang/ENIDrift.

Wolf, M., Tritscher, J., Landes, D., Hotho, A., Schlör, D., 2024. Benchmarking of synthetic network data: Reviewing challenges and approaches. Comput. Secur. 145, 103993. http://dx.doi.org/10.1016/j.cose.2024.103993.

Xu, J., Fan, J., Ammar, M., Moon, S., 2002. Prefix-preserving IP address anonymization: measurement-based security evaluation and a new cryptography-based scheme. In: 10th IEEE International Conference on Network Protocols, 2002. Proceedings.. pp. 280–289. http://dx.doi.org/10.1109/ICNP.2002.1181415.

Xu, S., Marwah, M., Arlitt, M., Ramakrishnan, N., 2021. STAN: Synthetic Network Traffic Generation with Generative Neural Models. In: Wang, G., Ciptadi, A., Ahmadzadeh, A. (Eds.), Deployable Machine Learning for Security Defense. Springer International Publishing, Cham, pp. 3–29. http://dx.doi.org/10.1007/978-3-030-87839-9_1.

Yang, Z., Liu, X., Li, T., Di Wu, Wang, J., Zhao, Y., Han, H., 2022. A systematic literature review of methods and datasets for anomaly-based network intrusion detection. Comput. Secur. 116, 102675. http://dx.doi.org/10.1016/j.cose.2022.102675.

Yang, L., Zhang, H., Shen, H., Huang, X., Zhou, X., Rong, G., Shao, D., 2021. Quality assessment in systematic literature reviews: A software engineering perspective. Inf. Softw. Technol. 130, 106397. http://dx.doi.org/10.1016/j.infsof.2020.106397.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., Yang, M.-H., 2023. Diffusion models: A comprehensive survey of methods and applications. ACM Comput. Surv. 56 (4), http://dx.doi.org/10.1145/3626235.

Yin, Y., Jang-Jaccard, J., Xu, W., Singh, A., Zhu, J., Sabrina, F., Kwak, J., 2023. IGRF-RFE: a hybrid feature selection method for MLP-based network intrusion detection on UNSW-NB15 dataset. J. Big Data 10 (1), 15. http://dx.doi.org/10.1186/s40537-023-00694-8.

Zha, D., Bhat, Z.P., Lai, K.-H., Yang, F., Jiang, Z., Zhong, S., Hu, X., 2025. Data-centric artificial intelligence: A survey. ACM Comput. Surv. 57 (5), http://dx.doi.org/10.1145/3711118.

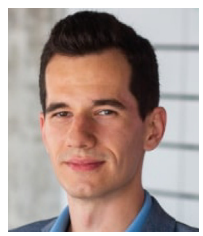
Zhang, C., Jia, D., Wang, L., Wang, W., Liu, F., Yang, A., 2022. Comparative research on network intrusion detection methods based on machine learning. Comput. Secur. 121, 102861. http://dx.doi.org/10.1016/j.cose.2022.102861.

Zhang, J., Wu, X., Sheng, V.S., 2016. Learning from crowdsourced labeled data: a survey. Artif. Intell. Rev. 46 (4), 543–576. http://dx.doi.org/10.1007/s10462-016-9491-9.

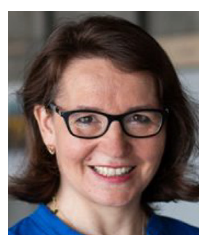
Zhang, C., Xie, Y., Bai, H., Yu, B., Li, W., Gao, Y., 2021. A survey on federated learning. Knowl.-Based Syst. 216, 106775. http://dx.doi.org/10.1016/j.knosys.2021.106775.

Zhou, D., Yan, Z., Fu, Y., Yao, Z., 2018. A survey on network data collection. J. Netw. Comput. Appl. 116, 9–23. http://dx.doi.org/10.1016/j.jnca.2018.05.004.

Zolanvari, M., Teixeira, M.A., Gupta, L., Khan, K.M., Jain, R., 2021. WUSTL-IIOT-2021 dataset for IIoT cybersecurity research. http://dx.doi.org/10.21227/yftq-n229, Dataset available at: https://www.cse.wustl.edu/~jain/iiot2/index.html.



Patrik Goldschmidt is a Ph.D. candidate at the Faculty of Information Technology (FIT) at Brno University of Technology (BUT), Czechia and a research assistant at Kempelen Institute of Intelligent Technologies (KInIT), Slovakia. With an interest in cybersecurity and computer networking since high school, he graduated with Bachelor's and Master's degrees from FIT BUT, focusing on DDoS detection and mitigation. In his Ph.D., Patrik studies the application of artificial intelligence and machine learning for cybersecurity, notably Network Intrusion Detection Systems, to enhance their performance, robustness, and trustworthiness while aiming to bridge the gap between academia and practice.



Daniela Chudá is an associate professor working at Faculty of Electrical Engineering and Information Technology (FEI), Slovak University of Technology (STU) and as an expert researcher and a Ph.D. supervisor at the Kempelen Institute of Intelligent Technologies (KInIT). She has authored or co-authored more than 60 publications in scientific journals and conferences, and regularly reviews submissions for international conferences and scientific journals. Her research is focused on information security and privacy, in particular behavioral biometrics in the context of user authentication, malware analysis and detection, network anomaly detection, phishing and malicious behavior.