Computer Networks and Communications
http://ojs.wiserpub.com/index.php/CNC/

UNIVERSAL WISER
PUBLISHER

Review

# Critical Review of Network Intrusion Detection Benchmark Datasets for Practical IoT Security

**Oluwasegun Apejoye**[*], **Nemitari Ajienka**, **Jun He**, **Xiaoqi Ma**

Department of Computer Science, Nottingham Trent University, Nottingham, NG11 8PT, United Kingdom
E-mail: n0966184@my.ntu.ac.uk

**Received:** 15 May 2025; **Revised:** 29 July 2025; **Accepted:** 4 August 2025

**Abstract:** The rapid expansion of Internet of Things (IoT) devices in modern environments introduces significant security vulnerabilities, increasing the risk of cyberattacks and potential physical threats to users. While Network Intrusion Detection Systems based on Machine Learning (ML-based NIDS) hold promise for mitigating these risks, the effectiveness of such systems is heavily influenced by the quality and representativeness of the datasets used for their development and evaluation. This study provides a critical review of publicly available benchmarking datasets commonly used to train ML-based NIDS for IoT security, with a particular focus on two pivotal but often overlooked factors: testbed configurations and feature extraction methods. The study's findings reveal critical inconsistencies in dataset design and feature sets, posing challenges to model generalizability and its real-world application. To address these issues, the study proposes clear criteria for dataset assessment and practical recommendations for researchers and dataset developers. The findings demonstrate the urgent need for standardisation to enhance reproducibility, enable fair comparison of intrusion detection models, and bridge the gap between academic research and practical IoT security solutions.

***Keywords***: benchmark datasets, feature extraction, Internet of Things (IoT) security, Machine Learning (ML), Network Intrusion Detection Systems (NIDS), reproducibility

## 1. Introduction

The influence of Internet of Things (IoT) innovations on daily human life is clear and undeniable. With advanced IoT technologies integrated into our society, particularly in our homes, the social dimensions of our lives are now deeply connected to this technology [1]. The rapid growth of these IoT devices has transformed various sectors, from smart homes to industrial automation, offering significant potential for enhanced convenience and efficiency. According to Statista [2], the number of IoT devices worldwide is projected to be over 32.1 billion by 2030, almost twice the 15.9 billion devices recorded in 2023. This market is expected to be primarily driven by consumer products, accounting for 60% of all IoT devices in 2023.

However, the proliferation of IoT devices presents significant security challenges. These devices routinely collect and process sensitive data while operating with constrained computational resources and limited security mechanisms, rendering them vulnerable to sophisticated cyberattacks. Threat actors exploit IoT vulnerabilities for various malicious purposes, including ransomware deployment, data theft, cryptocurrency mining, and botnet creation [1, 3–6]. The escalating

DOI: https://doi.org/10.37256/cnc.3220257228

sophistication of these attacks necessitates the development of robust, automated security solutions such as Network Intrusion Detection Systems (NIDS).

NIDS enhanced with Machine Learning (ML) have emerged as a critical component in IoT security infrastructure [7]. The complexity of contemporary cyber threats has driven the transition from traditional rule-based systems to ML-based approaches capable of adaptive threat detection. The past decade has seen significant advancements in applying ML to IoT security, drawing from its success in fields like image processing and natural language processing. Also, the availability of public datasets such as BoT-IoT, N-BaIoT, and CICIoT2023 has enabled the training of high-performance models with leading frameworks reporting detection accuracies approaching 99% and false positive rates below 0.01% [8, 9].

### 1.1 *Research gap and objective*

Despite many studies reporting high-performance metrics for ML-based NIDS, a significant disparity persists between theoretical performance and practical deployment capabilities. This is often due to insufficient attention being given to the end-to-end implementation pipeline, which is essential for real-world applications. The effectiveness and practical application of ML-based NIDS are heavily influenced by the quality of the network traffic data and the pertinence of the features extracted from the raw network data used during the training phase. Although academic studies [10–12] have underscored this impact, yet discussions on the fundamental processes of feature generation from raw network traffic data remain limited in scholarly literature. This gap is further exacerbated by the absence of standardised feature extraction methodologies, hindering model reproducibility and cross-dataset validation—critical requirements for operational deployment.

This study aims to bridge this gap by providing a comprehensive analysis of IoT-focused NIDS datasets, with particular emphasis on the feature sets and extraction methodologies involved in their generation. Our goal is to determine whether these datasets are suitable for training models in practical machine learning workflows and to highlight how researchers can approach future work to ensure theoretical performance translates to practical application. The study objectives are to:

1. Identify and review publicly available IoT-focused NIDS benchmarking datasets.
2. Analyse the feature extraction methods used in these datasets.
3. Evaluate the reproducibility and practical applicability of these methods.
4. Assess the suitability of these datasets for training ML models that can be deployed in real-world environments.

This methodological approach enables us to identify current limitations in dataset creation and documentation while providing practical insights for bridging the gap between academic research and operational deployment in the NIDS domain.

The rest of this study is organised as follows: Section 2 reviews related work on IoT-focused NIDS research. Section 3 outlines the methodology, detailing the research questions, information sources, search strategies, and the process for selecting papers. Section 4 provides a comprehensive description of IoT-focused NIDS datasets. Section 5 presents survey findings, addressing tool variability, feature extraction tools, dataset sources, and reproducibility issues in NIDS benchmarking. Section 6 discusses key Strategies for effective dataset utilisation and model generalisation. Finally, Section 7 concludes the study, summarising key insights.

## 2. Related papers

Several survey papers have been published over the years that cover different aspects of NIDS application in IoT security and their applications. However, most of these studies, such as [13–16], focus mainly on NIDS architecture and its application, with little or no in-depth discussion on the benchmarking dataset used to evaluate this solution or the practicality of translating these theoretical solutions into a real-life environment. Other studies, such as [17], only provided a comprehensive overview of general (non-IoT) network traffic datasets.

Khraisat et al. [14] presented an in-depth analysis of IoT Intrusion Detection System (IDS) taxonomy, emphasising deployment strategies and validation methods. However, their discussion on commonly used datasets for IDS development was limited, with only one of the nine being contemporary and applicable to the IoT context. Furthermore, they did not discuss the feature extraction techniques used in the benchmark datasets for training the ML-based IDS.

Chaabouni et al. [16] covers some open-source tools used to implement NIDS, including NIDS datasets. However, as IoT-focused NIDS datasets were recently published, the study did not cover such datasets.

Notably, Haque et al. [18] offers one of the few surveys examining IoT-specific datasets, discussing trends in learning techniques and algorithm efficiencies. However, the discourse on feature extraction methodologies remains underexplored, highlighting the need for this study's focused examination.

To our knowledge, no thorough survey exists of feature extraction techniques and their associated feature sets for benchmarking datasets in IoT security. This study's findings will help researchers evaluate the efficacy and practicality of deriving similar features (as presented in the benchmarking data they trained their model with) from raw network traffic from another network within feasible time constraints. Ultimately, this will help bridge the gap between academic innovations and practical applications in the NIDS domain—a challenge that has been recurrently identified in ongoing discussions within the field [10].

## 3. Methodology

This research employs a systematic review methodology adhering to the Preferred Reporting Items for Systematic Reviews and Meta-Analyses (PRISMA) guidelines [19] to ensure transparency, reproducibility, and thorough coverage of relevant IoT-specific NIDS datasets. Acknowledging the inherent limitations of keyword-based searches—especially in capturing critical datasets often found in non-indexed repositories or primarily cited within niche communities—we supplemented the PRISMA framework with insights gained from preliminary research, which utilised backward snowballing [20]. This uses the reference list from CIC-IoT23 [21]. By integrating this snowballing technique, we identified eight additional well-established datasets, including IoT-23, N-BaIoT, and BoT-IoT, which may have been missed in traditional database searches. The integration of these complementary approaches ensures both methodological rigour and comprehensive dataset coverage. Figure 1 illustrates the comprehensive systematic review process implemented in this research. The remainder of this section provides details about the primary methodology used in this study, the PRISMA methodology.

### 3.1 *Research questions*

This study aims to provide answers to the following questions.

RQ1. What datasets are publicly available for evaluating ML-based NIDS solutions within IoT networks?

RQ2. What are the most common feature extraction techniques and tools used in these datasets?

RQ3. Are there specific challenges associated with integrating models trained on benchmark datasets into an end-to-end pipeline for real-world applications?

By understanding the strengths and weaknesses of different feature extraction techniques, researchers and practitioners can develop more robust and effective NIDS solutions. This research employs a systematic approach to gather relevant articles in the literature that address research question 1. The insights derived from these studies subsequently inform the answers to the other research questions.

### 3.2 *Information source*

When conducting a systematic review such as this, the choice of database is crucial to ensure comprehensive coverage, high-quality sources, and efficient retrieval of relevant literature. Scopus stands out as an optimal choice for several reasons, particularly when compared to individual databases such as ACM Digital Library, Elsevier, IEEE Xplore, ScienceDirect, and SpringerLink, and other databases such as Google Scholar and Web of Science

Scopus database was used in this research study because of its extensive coverage, inclusion of multiple publishers, advanced search capabilities, and high-quality content. Unlike individual databases that are often limited to the publications of a single publisher, Scopus aggregates content from multiple publishers. This includes major sources such as Elsevier, Springer, IEEE, ACM, among others. Also, the content indexed in Scopus is predominantly peer-reviewed, ensuring that the literature sourced from it is of high quality and credibility.

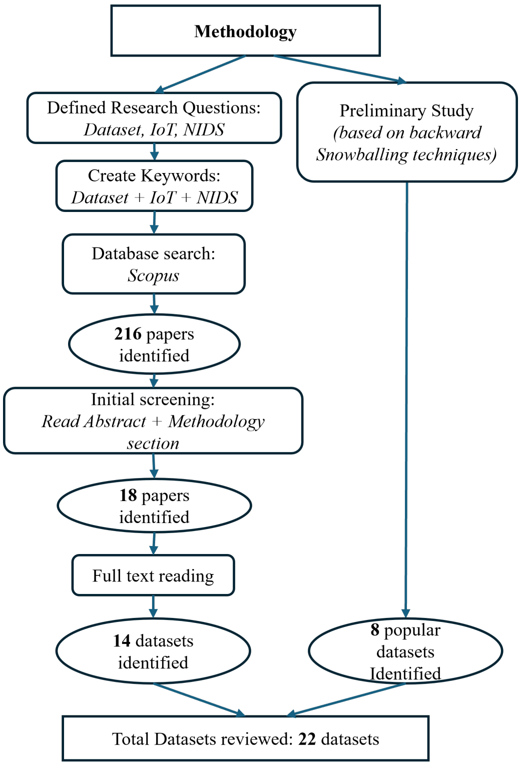
Methodology
Defined Research Questions: Dataset, IoT, NIDS
Preliminary Study (based on backward Snowballing techniques)
Create Keywords: Dataset + IoT + NIDS
Database search: Scopus
216 papers identified
Initial screening: Read Abstract + Methodology section
18 papers identified
Full text reading
14 datasets identified
8 popular datasets Identified
Total Datasets reviewed: 22 datasets

**Figure 1.** System design of mobile wireless sensor networks

## 3.3 *Search string*

The search term focuses on the intersection of datasets, IoT technologies, and NIDS. A comprehensive exploration of various terminologies associated with these topics was conducted, as detailed in Table 1, which outlines the full range of search terms employed in this investigation. The keywords were combined with Boolean operators to form appropriate search queries to retrieve relevant papers.

**Table 1.** Search string used in this study

| Search query | Keywords | Article section | Search criteria | # Result |
|---|---|---|---|---|
| dataset* AND ( IoT OR ( internet AND of AND thing* ) ) AND ( ( network AND intrusion ) OR ( network AND intrusion AND detection AND system* ) OR ( network AND intrusion AND system* ) ) | Dataset and IoT and NIDS | Title or Keywords | Year-2014–2024 | 216 |
| | | | Article type-Journal or Conference Paper | Document Type |
| | | | Publication-stage—Final | Conference papers-123 |
| | | | Language—English Only | Journal Articles—93 |

### 3.4 *Scope definition-inclusion/exclusion criterion*

This survey examines studies that published one of the following types of data:

- New raw network traffic data in Packet Capture (PCAP) format.
- Comma-Separated Values (CSV) format data with pre-extracted features derived from new or existing raw PCAP files.

The testbed for capturing the raw network traffic data must include IoT devices or be specifically designed for IoT networks. To capture the latest advancements in IoT security solutions, only datasets released between 2014 and 2024 are included.

Studies that generate non-IoT datasets or capture their data from general-purpose networks are excluded to maintain focus on IoT-specific data. Additionally, papers evaluating existing NIDS datasets are not included.

These criteria ensure that this study focuses on IoT-related NIDS datasets, effectively addresses its research questions, guarantees the accuracy and significance of the findings, and offers valuable insights into IoT-specific data and its applications.

### 3.5 *Paper selection process*

A total of 216 papers were identified through a database search using the keywords listed in Table 1. We initially screened the papers by reviewing their titles, abstracts, and methodology sections to exclude those not meeting our inclusion criteria described in section 3.4. This initial screening resulted in 18 papers that either generated new IoT NIDS datasets or enhanced existing ones. We then conducted a full-text review to ensure complete adherence to the inclusion criteria, which led to the selection of 14 papers and excluded 4 papers whose dataset are not accessible publicly or have been deleted. The 14 selected papers were combined with the 8 well-known datasets identified in our preliminary study, resulting in a total of 22 datasets. We analysed these 22 datasets in this study to identify trends, challenges, and advancements in NIDS datasets and feature extraction techniques for IoT security.

## 4. Description of IoT-focused NIDS datasets

As with any ML-based application which requires the availability of high-quality datasets for training the model, quality data is also critical for the training and evaluation of ML-based NIDS. In the evolving landscape of IoT security and NIDS, researchers have become fundamentally reliant on carefully engineered benchmark datasets for developing their ML-based systems. The justification for this trend is multifaceted, encompassing ease of prototyping, legal and privacy concerns associated with capturing real network traffic, issues related to labelling training datasets, comprehensive coverage of the attack space, reproducibility, and community collaboration. In this section, we provide a detailed description of publicly available datasets for training ML-based NIDS models.

### 4.1 *N-BaIoT (2018)*

Meidan et al. [8] from Ben-Gurion University of the Negev, Isreal, released the N-BaIoT dataset in 2018. It was specifically captured to evaluate their autoencoder-based NIDS system designed for network-based botnet detection in an IoT environment.

The raw network traffic was captured using Wireshark from a small-scale network with physical commercial IoT devices connected via WiFi and several access points or wired directly to a switch and a router. The network traffic flow of each IoT device was captured in isolation. The normal traffic is collected immediately following the device's installation in the network to capture the benign flows. To capture the attack traffic, the IoT devices were infected with two prevalent IoT botnet malware, Mirai and BASHLITE. The network traffic was captured with Wireshark in PCAP format.

To process the raw PCAP into an ML-friendly format, they extracted packet-level features in line with the approach employed in AfterImage [11]. Both the raw PCAP capture, and the pre-processed CSV formats are publicly available at [22]. All the CSV files combined contain 7,062,606 instances and 115 features.

The authors used this dataset to evaluate their anomaly detector, which is based on autoencoder. To do this, they split the raw traffic data into train, optimisation and test sets. They recorded promising 100% TPR and 0.007±0.01 FPR results and detected most attacks in less than a second.

The dataset's main limitation is its narrow focus on Mirai and BASHLITE botnets, which may bias the trained model and hinder its ability to generalise to other attack patterns. Additionally, processing the considerable number of features (115) can be computationally intensive, posing challenges for real-time detection and scalability in large-scale IoT deployments.

## 4.2 *Kitsune dataset (2018)*

This dataset was generated by researchers from Ben-Gurion University of the Negev, Israel, in 2018 to evaluate their proposed unsupervised autoencoders-based NIDS, Kitsune [11]. The dataset combines data captured from two different testbeds. One contained 8 surveillance cameras, and the other contained 9 IoT devices plus 3 PCs.

In the formal network, nine attacks, categorised into four types–MiTM (Man-in-the-Middle), DoS (Denial-of-Service), Reconnaissance, and Mirai malware attacks – were launched against the IoT cameras. In the latter testbed network, one of the security cameras was infected with a real sample of the Mirai botnet malware. The network traffics for each of the attack and benign scenarios were captured separately. For each dataset, clean network traffic was captured for the first 1 million packets, then the cyber-attack was performed.

To generate the ML-friendly CSV format, the authors extracted 115 packet-level features using their self-developed feature extractor tool, AfterImage, available on GitHub [23]. AfterImage works by extracting 23 features from five different time windows (100 ms, 500 ms, 1.5 seconds, 10 seconds, and 1 minute), totalling 115 features, whenever a new packet arrives in the network. The 23 features extracted from each time window contain statistical summaries such as weight, mean, standard deviation, etc., aiming to capture the temporal context of the packet's channel and senders. However, not all 115 features are captured for packets that don't contain Transmission Control Protocol (TCP)/User Datagram Protoco (UDP) datagram. The tool has a small memory footprint since the statistics are updated incrementally over damped windows. The labelling file, the pre-processed CSV and the original PCAP files are available on Kaggle [24].

The dataset was used to evaluate Kitsune, yielding good results. However, the author did not investigate cross-evaluation of their system with other datasets.

## 4.3 *BoT-IoT (2019)*

Koroniotis et al. [25] from the Cyber Range Lab, University of New South Wales Canberra, Australia (UNSW), captured and released the BoT-IoT dataset in 2018 and 2019, respectively. This dataset provides comprehensive network information with accurate labelling to help detect botnet attacks in IoT networks. It ushered in the era of IoT-focused NIDS datasets.

The testbed comprises a virtualised smart home network featuring five simulated IoT devices, instantiated using the Node-RED platform and employing the Message Queuing Telemetry Transport (MQTT) protocol for communication. To conduct various attack simulations, four Kali Linux Virtual Machines (VM) were deployed to execute Distributed Denial of Service (DDoS), Denial of Service (DoS), information theft, and information gathering attacks on the IoT devices. The Ostinato software was used to generate the benign network traffic to mimic traffic patterns representative of a real-world network environment. Network traffic corresponding to each attack vector and benign scenario was captured independently in PCAP format utilising the Tshark tool, to facilitate easy labelling.

The *Argus tool* was used to extract the dataset's original 29 flow-level features. Using SQL stored procedures, 14 statistical flow-level features—over a sliding window of 100 connections—were also extracted. To correctly assign the three label features—binary, attack categories, and subcategories—to the network flows, packets exchanged between the IP addresses of the attacker and victim machines were classified as attack packets.

The raw PCAP files contain a total of over 72 million records, and the extracted flow traffic in CSV is about 16.7 GB. To provide a more manageable size, the authors proposed 10 features (using Correlation Coefficient and Entropy metrics), which they argue are best for building an ML model and released a 5% subset of the entire dataset with the 10 best features and about 3.5 million instances. The raw PCAP, the argus files and csv files are available to download at [26].

The authors had no IoT-focused dataset to cross-evaluate their dataset with, so they evaluated it using a train/test splitting approach with three ML/DL algorithms.

A significant challenge in achieving reproducibility with this dataset stems from the unavailability of the script used for generating the additional features. This lack of documentation may hinder efforts to replicate similar features in different network environments. Additionally, the experimental setup did not utilise any actual IoT hardware, resulting in data that lacks the authenticity and variability of real IoT traffic. Moreover, the 5% subset is highly imbalanced, containing only 477 benign flows (0.01% of the total), which presents limitations for accurately modelling the normal network behaviour.

## 4.4 *IoT-SH (2019)*

Researchers at Cardiff University, UK, purposefully generated this dataset to evaluate their three-layer lightweight IDS architecture, which includes device identification, attack vs benign classification and attack identification [27].

The testbed consists of 8 physical IoT devices, including cameras, plugs, TVs, hubs and sensors. The IoT devices use one or more of the following protocols: Wi-Fi, BLE, and Zigbee. A laptop was used to launch four attack types—DoS, MITM/spoofing, reconnaissance, and replay—and capture the network traffic using tcpdump. Randomness was applied when launching the attacks to simulate a real adversary that might target the devices. Data collection ran for three weeks (benign) and two weeks (attack).

To extract features from the PCAP files for an ML application, the PCAP files were converted into Packet Description Markup Language (PDML) using Tshark. The PDML format provides access to all packet attributes that can be utilised as features. Therefore, 121 packet-level features were extracted to represent each packet using a bespoke Python script available on GitHub [28]. To accurately label the attack packets, packets with the attacker's MAC address and a timestamp within the attack timeframe were identified as attack packets.

The dataset is not publicly available but can be obtained upon request from the corresponding author. The authors utilised the dataset to evaluate their three-layer Intrusion Detection System (IDS). For this task, fewer than 10% of the 121 features were deemed sufficient to train machine learning models. Using a train/test/validate dataset splitting and 10-fold cross-validation, they recorded the following performance in F-measure, with 90% and 98.0% on binary and multiclass classification of the attacks, respectively.

## 4.5 *IoT-23 (2020)*

The dataset was compiled by researchers at the Stratosphere Laboratory, AIC group, FEL, CTU University, Czech Republic, from 2018 to 2019 and made available in 2020 [29]. Its primary objective is to provide a comprehensive collection of real, labelled IoT malware infections alongside benign IoT traffic, facilitating the development of machine learning algorithms in the field.

The testbed comprised three physical IoT devices, Somfy door lock, Philips Hue and Amazon Echo, used to generate the 3 benign network traffic captures, one from each device. Various malware samples were executed in a Raspberry Pi for the malicious scenario to capture 20 malware captures. The attack types contained in the dataset are PartOfAHorizontalPortScan, Mirai, Okiru, DDoS, Torii, C&C, Attack, File Download, and Heartbeat. The network traffic was captured in PCAP format.

The researchers utilised the Zeek tool to extract 21 flow-level features from the PCAP files, generating a log file in .text format; however, three of these features are redundant. Each log file was manually labelled (binary and multiclass) after analysing its corresponding PCAP file, ensuring an accurate representation of the traffic characteristics. The PCAP, log, and other documentation files are available publicly at the Stratosphere Lab webpage [30]. The author did not validate or cross-evaluate the efficacy of their dataset for developing an ML-based NIDS. However, this dataset remains one of the most widely used by researchers.

## 4.6 *IoTID20 (2020)*

Ullah and Mahmoud from Ontario Tech University, Canada, introduced the IoTID20 dataset in 2020 to provide a comprehensive dataset for developing and evaluating Intrusion Detection Systems (IDS) in IoT networks [31].

The dataset is a flow-based (CSV) dataset generated from the PCAP file of [32]. The original data source's testbed consisted of two physical smart home devices (speaker and camera), which were subjected to four different attacks (DoS, Mirai, MITM, and Scan). Other devices in the testbed included laptops, tablets, and smartphones. The attack was simulated using the Nmap tool.

To generate the dataset, Ullah and Mahmoud used CICflowmeter tool [33] to extract 80 features from the raw packet-based (PCAP) files [32] to generate the flow-based dataset (CSV format) for the IoTID20 dataset. Three additional label features were added (binary, category, and subcategory) to the dataset. The dataset is available at [34]

The author did not cross-evaluate their dataset with any existing dataset, but they evaluated the dataset using a train/test split approach and recorded some promising results. Of the 80 extracted features, when assessing the efficacy of their dataset for building the NIDS model, the author used 12 features which were at least 70% correlated with other features within the dataset.

While the dataset offers a realistic environment and a variety of attack types, it has some limitations that impact its effectiveness for training robust IDS models. One major critique is the lack of authenticity in generating the Mirai botnet attack packets. These packets were generated on a laptop and then altered to make it look like they came from an IoT device [32]. This might not accurately reflect the behaviour and traits of real IoT device traffic. As a result, there could be differences between the dataset and real-world situations, which could impact the performance of NIDS trained on this data.

Additionally, the dataset's limited device diversity poses a challenge. With traffic primarily from two smart home devices, the dataset does not fully represent the wide range of IoT devices in real-world environments. This limitation can hinder the ability of IDS models to generalise across different types of IoT devices and attack vectors. Despite these shortcomings, the IoTID20 dataset remains a valuable resource for academic research, providing a foundation for developing new intrusion detection techniques and highlighting areas for future improvement.

## 4.7 *ToN_IoT (2020)*

ToN_IoT is a dataset released in 2019 by researchers at UNSW Canberra's Cyber Range and IoT Labs in an effort to provide a rich variety of heterogeneous data sources for evaluating AI-driven IoT security solutions. It contains data from various protocols, including TCP, UDP, and MQTT [35–37].

The dataset integrates data from four heterogeneous sources: IoT network traffic (PCAP files), telemetry data from IoT/IIoT sensors, and operating system logs from Windows and Linux Operating Systems (OS). These data were collected across all four sources in parallel, creating a multifaceted dataset suitable for developing federated security systems.

The dataset's testbed mimics a large-scale Industry 4.0 network by interplaying edge, fog and cloud layers, deployed using multiple virtual machines running Windows, Linux, and Kali OS, along with seven simulated IoT devices. Normal traffic was generated using the Ostinato traffic generator, while a Kali VM was used to execute nine types of attacks, including scanning, backdoor, ransomware, DoS, DDoS, MITM, data injection, cross-site scripting (XSS) and password violations, against the IoT nodes.

The network traffic data, the source of interest in this review paper, comprises PCAP and CSV files. Each attack and benign scenario was captured separately in PCAP using the netsniff-ng tool. The Zeek tool was further utilised for feature extraction, yielding 44 flow-level features saved in CSV files. These features were designed to encapsulate information on connection identifiers, statistical summaries of flows, user attributes (about DNS, HTTP, and SSL activities), and violation attributes. The extracted features were drawn from various log files produced by Zeek, such as conn.log, http.log, weird.log, etc. Feature labels (binary and multiclass) were assigned based on timestamps correlating to attack events that compromised vulnerable nodes. Labels—binary and multi-class—were assigned based on timestamps corresponding to attack events targeting vulnerable nodes within the network.

The dataset, available at [38] in both PCAP and CSV formats, has over 22 million records across preprocessed CSV files, of which 3.56% are benign flows, and 96.44% are attack samples. A more manageable subset of 211,043 rows (train_test) is also accessible for training and testing machine learning models.

Although the authors did not cross-validate this dataset with existing ones, they employed a train/test split methodology and reported promising results. However, it should be noted that using features related to the device IP addresses and ports, which are specific to the testbed setup, could potentially inflate the model's performance metrics, which may not be achievable if the models are deployed in a different network configuration.

Moreover, a notable limitation of the dataset is its synthetic nature; the testbed did not involve actual users generating legitimate traffic, and most IoT/IIoT devices were simulated on a VM. This may affect the dataset's ability to accurately reflect real-world attack patterns, potentially hindering the generalisation capability of any models trained on it.

## 4.8 *MQTTset (2020)*

The dataset was generated in 2020 by [39]. It is focused on the MQTT protocol. The testbed is a simulated smart home environment where sensors retrieve information, such as temperature and humidity, at varying temporal intervals and send it to an MQTT broker. The 8 sensors in the testbed were simulated using IoT-flock, a network traffic generator tool, while the MQTT broker was based on Eclipse Mosquitto v1.6.2. Normal traffic was captured by simulating the normal sensor working mode, in which the sensors communicate with the MQTT broker. For the attack scenarios, the MQTT broker was targeted with a brute force attack, a malformed data attack and three DoS attack variants.

The normal and attack scenarios were captured separately in PCAP format using Wireshark. The Tshark tool was used to extract 33 packet-level features from the raw network traffic, generating a CSV format for ML training. Both the PCAP and CSV files are made available publicly on Kaggle [40].

The author did not cross-evaluate their dataset with any existing dataset; however, they evaluated the dataset using a train/test split approach and recorded some promising results. Nevertheless, some categories, such as Malformed data attack, were complex to detect.

## 4.9 *MQTT-IoT-IDS2020*

Given that MQTT is one of the most prominent IoT protocols and recognising the absence of a dedicated dataset focused on this protocol, Hindy et al. [41] developed this dataset through a simulated MQTT network architecture. This simulation serves as a viable alternative for researchers and practitioners requiring MQTT-specific data for their analyses and applications.

The testbed is a simulated MQTT network architecture consisting of 12 sensors and 1 camera, all simulated on a VM. The devices were subjected to 4 attack types–aggressive scan, UDP scan, Sparta SSH brute-force, and MQTT brute-force attack. Each scenario (attack and normal) is captured independently using tcpdump.

To generate ML-level CSV data from the PCAP files, PyShark and DPKT python libraries were used to extract three feature-level data formats– packet, unidirectional flow-based, and bidirectional flow-based—from the PCAP files. For the unidirectional format, 18 features were extracted using a custom script based on DPKT, which is available on GitHub [42]. A label feature was then added to categorise the attacks. Both the raw PCAP files and the CSV format of the three feature levels were made publicly accessible for researchers on IEEE DataPort [43].

However, a major issue with this dataset is that the existing documentation is not detailed enough. For example, since background normal operations were maintained during the attack scenarios, it is ambiguous how the normal activity was differentiated from attack traces during the labelling process.

The author did not cross-evaluate their dataset with an existing dataset but instead evaluated the dataset using a train/test split approach with each of the datasets (packet-based, bidirectional flow-based, and unidirectional flow-based). Their analysis indicated that models utilising bidirectional flow-based features achieved superior performance. Consequently, they concluded that flow-based features are more effective in distinguishing between benign traffic and MQTT-based attacks.

## 4.10 *MedBIoT (2020)*

Guerra-Manzanares et al. [44] from Tallinn University of Technology, Tallinn, Estonia, generated this dataset to contribute to existing datasets for training ML-based IDS for IoT networks with a special focus on botnet detection. The dataset focuses on the early stages of botnet deployment, specifically spreading and Command-and-Control (C&C) communication.

It was generated from a medium-sized IoT network comprising over 80 devices, organised into three interconnected segments: the internet, a monitoring, and an IoT LAN. Of the total devices, 80% were simulated, comprising various appliances such as locks, fans, and light bulbs, while three were physical devices, including one TP-Link light bulb. The dataset includes attacks from three well-known IoT botnets: Mirai, BashLite, and Torii, which were orchestrated against the IoT devices.

To capture baseline normal traffic, a Python script was developed to interact with the IoT devices, simulating typical user behaviours such as activating lights and adjusting fan speeds. The network traffic was recorded during these regular operational scenarios. For the attack scenarios, traffic was captured during the initial phases of the botnet propagation. Each normal and attack dataset was recorded separately in PCAP format utilising tcpdump.

To facilitate a machine learning-compatible data format, a custom script was developed to extract 100 distinct features from the raw PCAP files. This feature extraction methodology was guided by the work of [11]. Statistical features of network traffic were derived over five different time windows—100 ms, 500 ms, 1.5 s, 10 s, and 1 min—culminating in the creation of 100 features. The dataset is available in both PCAP and CSV formats for public access [45].

The authors performed an initial evaluation of the dataset utilising a train/test split and achieved promising results, including an F1 score of 0.97 for the random forest classifier. However, a significant limitation of this dataset is the unavailability of the feature extraction script, which hinders other researchers from accurately replicating the feature set in their environments. This lack of transparency could affect the practical deployability of models trained using this dataset.

## 4.11 *ECU-IoHT (2020)*

This dataset was generated in 2020 by [46] at Edith Cowan University, Australia, to help the healthcare security community analyse attack behaviour and develop robust countermeasures.

To generate the dataset, an IoMT testbed was developed to generate the dataset consisting of 3 IoMT sensors connected to the Libelium MySignals healthcare device, which in turn communicates the sensor data via WiFi to the Libelium cloud server. A Kali Linux VM running on a Windows PC was used to attack the IoMT setup. The attacks launched in the publicly available dataset include Smurf, DoS, ARP spoofing, and Reconnaissance attacks. The network traffic was captured using Wireshark.

To label the dataset, manual packet analysis was performed to differentiate between attack traffic and normal traffic in the capture file using the time of the attack launch. This approach differs from that of other authors, who generate separate PCAP files for each attack or normal scenario and label the flows accordingly. Flow-based data was generated (using Argus and Tsark from the raw PCAP file) for evaluation purposes but was not made publicly available. Instead, the packet-based PCAP file was directly converted to spreadsheet format (.xlsx) and made publicly available to researchers with the addition of two label features [47]. The dataset contains a total of 111207 packets, of which 21% are normal packets.

The authors used twelve different types of anomaly detection algorithms to evaluate the dataset. Their results show that most models performed poorly across most attack types. Moreover, the LDCOF that recorded the best performance failed to identify the DoS attacks, partly due to the small representation of the DoS attack in the dataset (639 instances).

Notably, the format in which this dataset was made available Excel Spreadsheet (XLS) is not suitable for researchers, as features can't be extracted from it, and the current information does not contain extracted features or sufficient details to perform any machine learning. Additionally, this dataset differs from typical DoS scenarios, where large network traces are often left. This discrepancy raises questions about whether the data was correctly configured and captured.

## 4.12 *TCP FIN flood and Zbassocflood dataset (2021)*

To fill the gap in the lack of datasets that used real-life devices and diverse communication protocols, Stiawan et al. [48] from the Computer Network and Information Security (COMNETS) laboratory at Sriwijaya University, Indonesia, generated the TCP FIN Flood and Zbassocflood Dataset and made it public in 2021 [49].

The testbed utilised a real-life star network topology, comprising a wireless router that connected six physical sensor nodes to a server (a PC). Four sensor nodes communicate via WiFi and other two via ZigBee protocol. The sensors capture the following data: humidity, gas, fire, soil moisture, water level, and temperature. The Wi-Fi-based nodes send their data directly to the server, while the ZigBee-communicating nodes first broadcast their data to the Raspberry Pi middleware, which then sends the data to the server via Wi-Fi. The testbed focused on three DoS/DDoS attacks: a "finish" (FIN) flood and a User Datagram Protocol (UDP) flood targeting WiFi communication (Nodes 1-4, the Middleware, and the server), and a Zbassocflood/association flood targeting ZigBee communication (Nodes 5–6).

For data capture, Wireshark was connected to the router to capture the network traffic of the six IoT nodes. Additionally, an AVR RZUBSSTICK was connected to Nodes 5 and 6 (ZigBee nodes) to capture the dedicated ZigBee traffic. Wireshark was also installed on the server PC to capture the non-IoT traffic on the server side. The resulting overall dataset is labelled accordingly as normal or attack. Feature extraction was performed using the Tshark tool, resulting in 95 features for the WiFi dataset and 64 for the Zigbee dataset. However, only the raw PCAP files were made publicly available to the research community [49].

The author did not perform a cross-evaluation of their dataset with existing dataset. However, they evaluated the dataset's robustness for recognising the types of data traffic, using a rule-based signature analysis method that used Naïve String Matching. They utilized the FIN and Zbassocflood-related datasets for evaluation and achieved promising results, precisely an accuracy level of 99.92%, a Precision of 100%, a False Positive Rate (FPR) of 0, and a False Negative Rate (FNR) of 0.0869.

## 4.13 *Edge-IIoTSet (2022)*

Ferrag et al. [50] generated the Edge-IIoTSet in 2022 to add to the limited number of IoT/IIoT datasets available for cybersecurity research. In their work, they propose a new IoT /IIoT dataset collected from a sophisticated seven-layer testbed, including more than 10 IoT/IIoT devices and with 14 IoT and IIoT protocol-related attacks lunched in the testbed.

The testbed is a seven-layer testbed IoT/IIoT environment featuring a diverse array of physical devices, sensors, communication protocols, and versatile cloud/edge configurations. The testbed consists of over 10 IoT/IIoT devices.14 attack types categorised into five threats were launched against the IoT devices.

The network traffic was captured in PCAP form using Wireshark. To provide a flow-level dataset for training ML algorithms, Zeek and TShark tools were used to extract 1176 features from the raw PCAP files (network traffic) and other sources, including system resources, logs, and alerts. A Python script based on the *Yellowbrick* package was then used to reduce the features based on their correlation to form the 61 features present in the dataset. Additional label features were also added for binary and multiclass classification. Both the raw PCAP and pre-processed CSV files are made available to the research community as an open dataset on IEEE DataPort [50].

The author did not perform a cross-evaluation of their dataset with the existing dataset, as done by the IDSAI dataset authors. However, they evaluated their dataset suitability for training ML-based NIDS by assessing the performance of 5 ML/DL algorithms using federated and centralised learning approach. For the centralised mode and using 2-hidden layer DNN, they recorded accuracy of 99.99%, 96.01%, and 94% for the 2-class, 6-class, and 15-class classification tasks, respectively.

## 4.14 *CIC IoT dataset (2022)*

In 2022, the CIC IoT dataset was developed by Dadkhah and colleagues at the Canadian Institute of Cybersecurity (CIC), Faculty of Computer Science, University of New Brunswick (UNB), specifically designed for the profiling, behavioural analysis, and vulnerability testing of various IoT devices [51].

The testbed comprises 60 distinct physical IoT devices, enabling the accurate simulation of a real-world smart home environment. This includes devices utilizing WiFi, ZigBee, and Z-Wave protocols, such as cameras, a lamp, and a coffee maker. Although the primary focus of the dataset is on the behavioural analysis of IoT devices in both idle and powered-on states, the researchers also executed DoS and RTSP brute-force attacks on ten selected IoT devices to capture resultant traffic. This makes the dataset suitable for building NIDS.

Network traffic was captured using Wireshark in raw PCAP format. Subsequently, flow-level datasets were generated employing the Python packet manipulation library, DPKT, which facilitated the extraction of 48 features from each device's network captures. Both the raw PCAP files and the resulting CSV format dataset are accessible on the CIC dataset page [52].

It is essential to note that because the dataset is primarily oriented towards device profiling, the authors did not investigate its potential for constructing ML-based NIDS. The custom tool employed for feature extraction used in the dataset generation, combined with inadequate documentation of the extraction methodologies and the absence of the source code, may hinder the reproducibility of this process by other researchers and, therefore, limit the deployability of models trained with this dataset.

## 4.15 *NF-BoT-IoT-v2 and NF-ToN-IoT-v2 (2022)*

To promote a standardised feature set between NIDS datasets and to facilitate a fair comparison of ML-based network traffic classifiers across different NIDS datasets, Sarhan et al. [10] provided two NetFlow-based datasets, NF-BoT-IoT-v2 and NF-ToN-IoT-v2. These are extended versions of two well-known IoT NIDS datasets, BoT-IoT and ToN-IoT.

To generate the dataset, they used the nProbe tool to extract 43 NetFlow version 9 features from the publicly available PCAP files of the respective dataset. To label the dataset, they match the five flow identifiers: source/destination IPs and ports and protocol to the ground truth attack events published by the original dataset's author. For the NF-BoT-IoT-v2 dataset, the total number of data flows is 37,763,497, out of which 37,628,460 (99.64%) are attack samples and 135,037 (0.36%) are benign. The dataset has four attack categories. NF-ToN-IoT-v2 contain a total number of 16,940,496 data flows out of which 10,841,027 (63.99%) are attack samples and 6,099,469 (36.01%) are benign. The data is available publicly at [53].

To assess the datasets, the researchers conducted a comparative analysis of classifier performance using both the extended NetFlow data and the original feature set. They observed a reduction in the False Alarm Rate (FAR) and prediction time when utilising their curated dataset, thereby validating the effectiveness of the extended NetFlow feature set. Notably, the feature extraction process implemented in this dataset is not only automated through nProbe but is also designed for reproducibility by other researchers in their network environments.

## 4.16 *AIoT-Sol (2022)*

Min et al. [54] developed the AIoT-Sol Dataset in 2022 to complement existing datasets by providing a comprehensive labelled dataset with more attack types that reflects real-world IoT attacks and can be used for developing ML based anomaly detection systems for IoT networks. They also aim to provide a systematic approach to design and develop an IoT testbed to generate a similar dataset.

The dataset testbed consists of six IoT devices: a Raspberry Pi (running an MQTT broker), a smart camera, a smart socket, a smart plug, a smartphone, and a smartwatch; a VM running four deliberately vulnerable applications; and a PC for the attack. The four deliberately vulnerable applications and the IoT devices are victims of 17 attack types from a PC. The 17 attack types launched were informed by the 2018 Open Web Application Security Project (OWASP) top ten IoT risks.

Benign traffic was captured for 2 hours while running the network devices in standby mode with normal user interaction. Each attack scenario was launched and captured with Wireshark at different times to allow for easy dataset labelling, a convention among dataset authors.

To extract the flow-based data, the CICFlowMeter was used to derive network statistical features and labels from the PCAP files. The public dataset is exclusively available in the CSV flow format, comprising 83 features. Additionally, it contains one label feature categorised into one benign and 16 different attack types. The dataset contains approximately 5.1 million total flows, of which around 2.4 million are benign flows.

A major concern with this dataset is the disproportionate representation of specific attack types. For instance, vulnerabilities like Server-side Request Forgery (1,756 instances), and Open Redirect (1,237 instances) are heavily underrepresented when compared to other attack types, such as SSDP flooding attack (970,418 instances) and MQTT brute force attack (482,553 instances). This uneven distribution will affect the performance of ML trained with this dataset in detecting the underrepresented attacks.

## 4.17 *LATAM DDoS-IoT (2022)*

The LATAM dataset was designed and created by collaborative effort between researchers at Aligo, Universidad de Antioquia, and Tecnologico de Monterrey [55]. It was generated to address the deficiency of datasets derived from testbeds that utilise real user traffic and launch DDoS/DoS attacks on physical IoT devices. The dataset aims to enhance the understanding of attack vectors and bolster defences in real-world IoT environments.

Its testbed consisted of 4 physical and one simulated IoT device deployed in an Software-Defined Network (SDN) environment. The IoT devices were the victim of protocol-based DDoS/DoS attacks launched by Kali VMs using Hping3 (UDP and TCP-based) and GoldenEye (HTTP-based) tools. Normal traffic was captured over 50 minutes using a gatherer node from a span port mirroring Aligo's customers' production activity. Each victim's attack time window was 10 minutes. Each attack and normal scenario was captured separately using the tcpdump tool, and the PCAP files were named according to the scenario for ground-truth labelling.

The Argus tool was used to parse each PCAP file into network flow (CSV) format, containing 18 features and two label features. The features extracted in the dataset were informed by some BoT-IoT features investigated in prior work of some of the dataset's authors [56]. To label the dataset, all flows from each file are labelled to reflect the scenario from which it is generated. The DDoS version (LATAM-DDoS-IoT) contains a large number of attack flows in the ratio of 1 normal to 61 attack traffic and a total of 49,666,991 flows with 20 attributes. The PCAP and network flow (CSV) are publicly available on IEEE DataPort [57]. The consideration of normal traffic from real users in the dataset allow for the modelling of real user traffic using the dataset.

To validate their dataset, the researchers developed a Java application that incorporates the entire ML-based NIDS pipeline. This application employs a decision tree model trained on the LATAM-Bot-DDoS-IoT dataset, a hybrid dataset comprising their LATAM dataset and a balanced variant of the BoT-IoT datasets from [56]. The application was designed to safeguard a simulated SDN architecture that integrates physical and virtual components against Denial-of-Service (DoS) and Distributed Denial-of-Service (DDoS) attacks. They achieved strong performance metrics, with 94.6% accuracy and a 95% F1 score, and zero false positives, while maintaining a false negative rate of 0.09%.

## 4.18 *CICIoT2023*

In 2023, Neto et al. [21] generated this robust dataset at the CIC Institute of UNB, with the aim of expanding the breadth of malware activity representation. This initiative is designed to enhance the development of security analytics applications tailored for real-world IoT environments, particularly within smart home contexts.

The testbed simulates a smart home environment and incorporates a diverse array of devices utilising WiFi, ZigBee, or Z-Wave protocols. The devices include smart home devices, cameras, sensors, and microcontrollers (Raspberry Pi units). Notably, this testbed comprises 105 physical IoT devices, marking it as the most extensive of all the datasets we reviewed. These devices were strategically categorized based on their vulnerability to various attack types; for instance, web-based attacks were directed at devices supporting web applications. A total of 33 distinct attack vectors were executed, classified into seven primary groups: DDoS, DoS, Recon, web-based, brute force, spoofing, and Mirai. All attacks involved malicious IoT devices targeting other IoT devices, contributing to one of the few datasets that detail both attacking and victimized IoT devices.

The benign traffic dataset was generated through 16 hours of transactional data from the IoT devices during both idle states and periods of human interaction. To extract features from the captured traffic, the DPKT Python library was used to create a custom script to extract 47 distinct features from the PCAP files. These features include summary statistics, such as mean values, calculated over fixed-size packet windows of 10 or 100 packets. This approach was employed to address discrepancies in data size, particularly relevant when comparing DoS attack traffic (100 packets per window) to Command Injection traffic (10 packets per window), thereby capturing the nuances of the packet sequences exchanged between the host devices. Extracted features are based on proposals from the creator of CIC IoT dataset 2022 and Edge-IIoTSet [50, 51]. The PCAP and CSV formats of the dataset are publicly available on the CIC dataset page [58]. To ensure the reproducibility of the feature extraction process by other researchers, the authors release sample source codes used in extracting the original features presented in the dataset.

The author did not cross-evaluate their dataset with any existing one, however, they adopted 5 ML algorithms to evaluate the dataset in binary, multi-class with 8 classes or multi-class with 34 classes classification problems using a train/test split approach. They recorded a good performance on binary classification, but the performance degraded slightly (0.96 to 0.71 - F1 score) on the multiclass classification problem, indicating the complexity of classifying many attack classes.

### 4.19 *IoMT-TrafficData (2023)*

In 2023, researchers at the Polytechnic of Leiria, Portugal, created the IoMT-Traffic dataset to capture and analyse network traffic in an IoMT environment. The dataset was created using a testbed that simulated a sports clinic environment comprising three main network areas: the Adversary Area, the General IoT Area, and the Wireless Body Area Networks (WBAN) [59]. The testbed comprises four IoT devices, including PIR motion sensors, DHT11 sensors, smartwatches, and heart rate belts, utilising communication protocols such as MQTT, CoAP (Constrained Application Protocol), Bluetooth, and Wi-Fi to reflect real-world usage scenarios.

Multiple cyberattacks are executed within the testbed to generate the attack data. These attacks include variations of DoS attacks (Apache Killer, R-U-Dead-Yet, Slow Read, and Slow Loris), ARP Spoofing, CAM Table Overflow, MQTT Malaria, Network Scanning, Bluetooth Reconnaissance, and Bluetooth Injection. The attacks target different devices and protocols within the testbed, providing diverse malicious traffic data.

Separate approaches were used to capture data from the two distinct protocols, IP-based and Bluetooth, represented in the testbed. Tcpdump was used to capture data for IP-based traffic, and a Bluetooth dongle was used for Bluetooth traffic. For the IP-based traffic, the benign traffic was collected over approximately 120 hours, resulting in around 3 million packets, while malicious traffic was collected over about 5 hours, generating over 15 million packets in total. The data was collected and stored in PCAP format with tcpdump. For the Bluetooth-based traffic, a Bluetooth dongle was used to capture over 974,630 packets, comprising approximately 90% normal traffic, 4% reconnaissance attacks, and 5% Bluetooth injection attacks.

Both packet-based and flow-based features were extracted from the raw PCAP of the IP-based traffic. To generate the flow-based features, proceeded the PCAP files with Zeek and combined the conn.log and flowmeter.log files generated by Zeek using the unique identifier, uid, to produce the 3.2 million flow-level data records, of which 10% are normal flows. The packet-based approach utilizes Tshark to extract detailed packet information from each packet, leveraging the authors' domain knowledge, and then converts the PCAP file into CSV format. The flow-based data consisted of 24 features (including 2 label features), while the packet-based data comprised 23 features. The data is made available to researchers in the original PCAP format and CSV format in three subsets (BLE, IP-based packet, and IP-based flow data) on Zenodo [60].

The authors' preliminary comparative analysis reveals that the flow-based approach offers superior outcomes compared to the packet-based approach. They also noted that using flows in an IDS can reduce data transfer volume by approximately 90% compared to packet-based data. The authors did not conduct a cross-evaluation with existing data.

### 4.20 *IDSAI (2023)*

Fernando et al. [61] generated the IDSAI dataset to provide a dataset for contrasting model generalisation across different network setups. The testbed comprises a network of sensors that gather various environmental parameters, including temperature, humidity, carbon monoxide, and ultraviolet intensity, which are subsequently relayed to a Raspberry Pi 3 node. This node aggregates and transmits the data to the cloud for further statistical analysis and visualisation. Within this framework, ten distinct types of cyberattacks were executed against the node, and the Wireshark tool was employed to capture the network traffic in PCAP format. Attacks launched include DDoS, brute force, MiTM, UDP port scan, TCP null and 5 variations of the DoS attacks.

For feature extraction, the researchers developed a proprietary script to extract 24 features from the raw traffic data, transforming it into a machine learning-friendly CSV format. These features were identified through a thorough analysis of the dataset and were partly informed by definitions from prior research. To address potential biases and data imbalance, the dataset was carefully curated to include an equal number of instances for each type of intrusion. Overall, the dataset consists of 1,000,000 samples—50,000 of which are normal traffic, alongside 5,000 samples corresponding to each of the ten attack categories. Only the CSV format of the dataset has been made publicly available on GitHub [62].

To validate the efficacy of the IDSAI dataset, the authors conducted a cross-evaluation by training models on the dataset and testing their performance against a portion of the raw PCAP data from the BoT-IoT dataset. Utilising the same in-house script for feature extraction, they extracted an identical feature set from the BoT-IoT raw PCAP data. The best-performing models, trained on the IDSAI dataset, successfully predicted binary labels (attack versus non-attack) on the BoT-IoT data, achieving over 90% accuracy, thereby underscoring the robustness of the dataset for training generalised NIDS models.

The IDSAI dataset has notable limitations, including insufficient documentation on the feature extraction process and the lack of public access to the feature extraction script. These affect the reproducibility of the feature extraction methodology and, ultimately, the viability of training a deployable NIDS model using this dataset. Additionally, the raw traffic data is not publicly available for researchers to engineer their own features.

### 4.21 *GRASEC-IoT (2024)*

Hamouche et al. [63] present GRASEC-IoT (a GRAph-based dataset for SECurity enforcement in IoT networks), a curated collection of graph data specifically designed for research into the application of Graph Neural Network algorithms for attack detection within IoT networks.

The dataset was generated in a simulated IoT network environment, which includes a user network, an adversarial network, and remote servers. The user network features VMs emulating end-user devices and IoT devices, while the remote servers host the two most common IoT communication protocols, CoAP and MQTT, which were set up to service IoT devices in the user network. The adversarial network utilizes a bot master (Kali VM) as a Command and Control (C&C) server to orchestrate attacks, including DDoS, brute force, and port scanning. The simulated IoT devices include MQTT-based sensors (light intensity, temperature, and smoke) and CoAP-based devices (door lock and fan speed controller).

Benign traffic was generated through a VM emulating user activities, and a VM running IoT-Flock software simulated IoT devices communicating with remote servers via MQTT or CoAP protocols. From the adversary network, malicious traffic was generated by launching 3 high-level attacks against the remote servers (which serve as the user network server). These attacks included DDoS attacks, brute-force attacks, and port scanning.

Network traffic was captured using Wireshark and saved in PCAP format. The CICFlowmeter tool was used to extract 83 features to generate the flow-based version of the dataset. This graph-based data can be used to train Graph Neural Network (GNN) models for IoT security. Both the JSON and CSV formats of the dataset can be accessed on Gitlab [64]. The flow-based dataset contains 740,000 flows with ground-truth labels. The JSON format can be used for training GNN and NN models.

The authors did not evaluate the dataset but presented the network traffic data in graph format to capture the interactions and relationships between devices, services, and communication patterns within the IoT network.

### 4.22 *CICIoMT2024*

The CICIoMT2024 dataset consists of Internet of Medical Things (IoMT) network traffic captured and released in 2024 by [65] from the CIC institute, UNB, to serve as a realistic benchmark dataset for evaluating IoMT security solutions.

The dataset testbed comprises 40 IoMT devices, consisting of 25 real devices and 15 simulated devices. The IoT devices utilise one of the popular IoT protocols commonly found in healthcare, including Wi-Fi, MQTT, or BLE. All the MQTT devices were simulated using the IoTFlock simulation tool, while the Bluetooth and WiFi-enabled devices were physical. The behavioural patterns of the IoT devices were captured by studying their power, idle, active, and interaction states. These traffics form the benign traffic presented in the dataset. For the attack traffics, the BLE devices were subjected to a DoS attack, while the WiFi and MQTT devices were victims of 18 distinct cyberattacks categorised into five: DDoS, DoS, Recon, MQTT, and spoofing attacks. The network traffic of both attack and normal scenarios was captured using Wireshark. However, the traffic data of the BLE devices were captured and grouped separately from the WiFi and MQTT data with the help of Ubertooth One sniffer.

The DPKT Python library was used to extract 39 features from the raw PCAP files to generate flow-level CSV data formats—a subset of features found in CIC_IoT23. The flow-based data contains 3 labelling categories (binary, 6 major categories and 19 sub-category attacks. The dataset is available on the CIC dataset page [66].

The authors employed a comparable methodology to assess their dataset and noted results that closely resembled those found in CICIOT2023. This similarity can be attributed, in part, to the fact that nearly the same group of researchers generated both datasets.

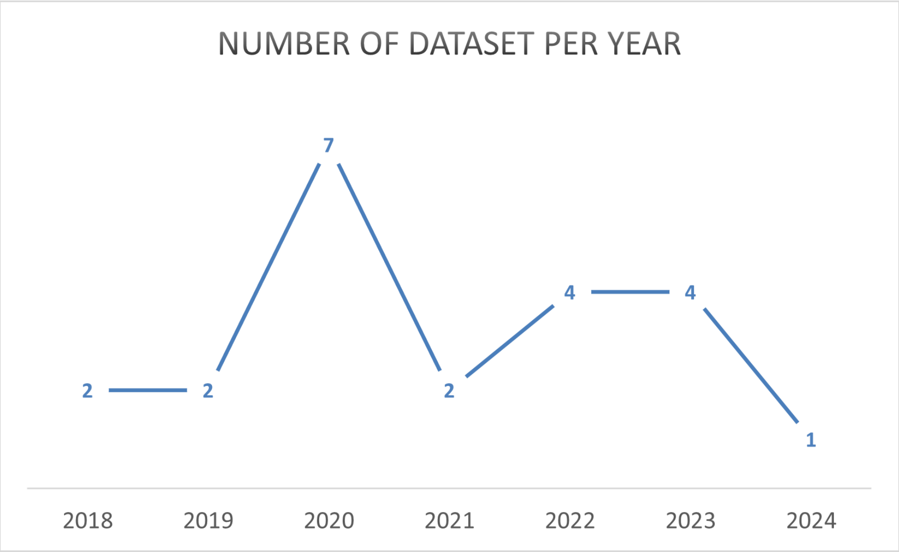

**Figure 2.** Number of IoT-focused NIDS datasets published by year

Figure 2 shows the number of IoT NIDS datasets published per year from 2018 to 2024. It hig the fluctuating publication trends in datasets, with no consistent growth.

# 5. Survey findings

## 5.1 *Tool variability and feature inconsistency in NIDS datasets*

Figure 3a-f presents a comparative analysis of feature set intersections among datasets utilising identical feature extraction tools. It highlights that, even when employing the same underlying tool (e.g., Argus), most datasets exhibit uniqueness in their feature representations. For example, the Bot-IoT and LATAM-DDoS datasets–both processed through Argus–capture fundamentally distinct aspects of network behaviour, as evidenced by their lack of overlapping features (Figure 3a,f)

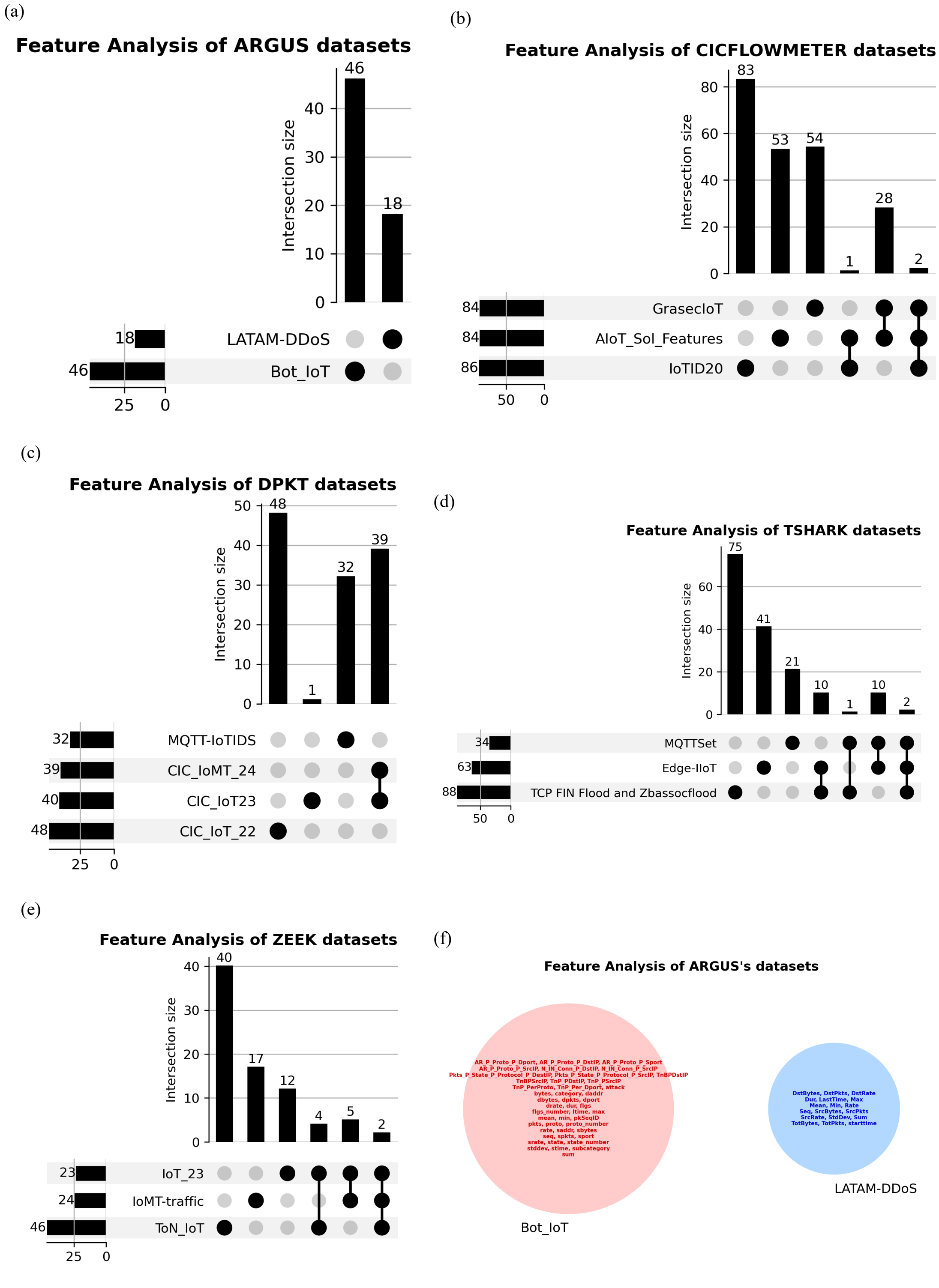

**Figure 3.** (a) Feature analysis of datasets that used Argus tool for feature extraction; (b) feature analysis of datasets that used CICFlowMeter tool for feature extraction; (c) feature analysis of datasets that used DPKT library for feature extraction; (d) feature analysis of datasets that used Tshark tool for feature extraction; (e) feature analysis of datasets that used Zeek tool for feature extraction; (f) feature analysis of datasets that used Argus tool for feature extraction

Furthermore, there are persistent inconsistencies in feature names across datasets derived from the same tool. The CICFlowMeter (Figure 3b) case acutely demonstrates this challenge, where semantically equivalent features appear as "Source IP" (AIoT_Sol), "Src_IP" (IoTID20), and "Src IP" (GrasecIoT). Such terminological discrepancies and unique feature sets complicate cross-dataset comparisons and model benchmarking, revealing the need to standardise features. These findings underscore that tool selection alone cannot ensure feature compatibility.

## 5.2 *Feature extraction tools*

Due to the nature of network traffic, useful features must be extracted from the raw network dataset before training ML algorithms. Otherwise, gaining useful insights from the data won't be easy. From the reviewed dataset in this study, several tools have been used for this process, with varying impacts on the resulting feature spaces. This section briefly discusses each of these tools.

### 5.2.1 *CICFlowMeter*

This open-source bidirectional network flow analyser extracts 80+ features (e.g., packet lengths, inter-arrival times, flags) from PCAPs, generating CSV outputs with flow identifiers (IPs, ports, protocol). It originally started as ISCXFlowMeter and was extensively discussed by the authors [33]. It was originally written in Java but now has a Python implementation. The Python version is available on GitHub [67]. Also, the original C software is on GitHub [68]. It was used in IoTID20, AIoT-Sol, and GraSec-IOT datasets.

### 5.2.2 *Tshark*

This is Wireshark's command-line tool for packet analysis. It can capture packet data from a live network or read packets from a previously saved capture file. It can extract detailed protocol-specific fields and statistics at the packet level. It is highly customisable via the display filter flag (-T fields), and this feature offers a lot of flexibility for users to extract relevant features that suit their needs. It is often used for parsing metadata (e.g., HTTP headers, DNS queries, TCP, etc) or exporting to JSON/CSV. It was used to generate the following datasets: Edge-IIoTSet (alongside Zeek), IoT-SH (alongside a bespoke python script), MQTTset and TCP FIN Flood and Zbassocflood.

### 5.2.3 *Zeek (formerly Bro)*

It is an open-source network security monitoring tool that analyses raw network traffic and generates several rich log files (e.g., conn.log for flows, http.log for HTTP traffic, and traffics from several other application layer protocols) from either a capture file or network interface(s) [69]. It can be run passively, making it ideal for real-time or offline PCAP analysis. It was used in Edge-IIoTSet (alongside Tshark), ToN-IoT, IoMT-Traffic, and IoT-23 datasets.

### 5.2.4 *nProbe*

It is a probing tool that can extract up to 365 NetFlow/IPFIX-based features [70], including throughput, jitter, and TCP metric from existing files and live capture. Its many extractable features offer users the flexibility to extract relevant features that meet their needs. It is a command-line tool available as both an open-source version, available on GitHub [71]and enterprise versions that can be purchased from the Ntop organisation. It was used by [10] in generating their datasets, NF-BoT-IoT-v2 and NF-ToN-IoT-v2.

### 5.2.5 *Argus*

It is one of the earliest works on netflow analysers that started in 1984. It can process raw packet data from either a file or network interface(s) and extract network flow-level features to generate bidirectional flow data in CSV or JSON format (Argus data). The Argus data can be made available for near real-time analytic processing or stored and used to generate a network activity audit information system [72]. It was used in BoT-IoT (alongside a custom script) and LATAM-DDoS-IoT. It is an open-source tool available on GitHub https://github.com/openargus/argus [72].

### 5.2.6 *DPKT*

This is a lightweight Python packet parsing library for manual feature extraction. It supports TCP/IP protocols and enables custom feature engineering (e.g., payload analysis, header fields). This library was used (alongside Pyspark) for the feature extraction in four datasets: MQTT-IoT-IDS2020, CICIoT2022, CICIoT2023, and CICIoMT2024. For more information, we refer readers to the project's GitHub page [73].

### 5.2.7 *AfterImage*

This is the feature extraction framework introduced by [11] and used in generating their dataset, the Kitsune dataset. Its variation was also used in generating the N-BaIoT dataset. It can extract up to 115 packet-level features from network capture by monitoring the pattern of each network channel through damped incremental statistics, generating a feature vector for every packet. This vector reflects the temporal context of the packet's channel and sender. Its source code is available on GitHub [23]

Each of the above tools suits different needs: CICFlowMeter for standardised features, Zeek/Tshark for deep protocol inspection, DPKT for custom scripting, and Argus/nProbe for flow exports.

## 5.3 *Dataset source*

Table 2 categorises the feature extraction tools outlined in Section 5.2 based on their core methodologies for analysing network traffic. These tools derive features from raw network data through two primary techniques: packet-level features and flow-level features. Below is a concise overview of these two data types.

**Table 2.** Data source generated by different feature extraction tools

| Data Source | Tool | Usage |
|---|---|---|
| Flow-based | CICFlowMeter, nProbe, Argus, Zeek, DPKT | 15 datasets |
| Packet-based | Tshark, AfterImage | 8 datasets |

***Packet-based data:*** This data source contains a Packet-level feature. Packet-level feature extraction represents each individual network packet as a distinct row in the parsed dataset (typically CSV format), preserving fine-grained protocol and payload characteristics. Common features in this category are features related to individual application protocols such as tcp.flags, frame.len, http.content_lenght, etc. This feature level captures atomic network events but may incur scalability challenges in high-throughput environments due to storage and processing overhead. A common tool for extracting this type of feature is Tshark. AfterImage [23] also extract this level of feature. This data source is presented in 8 datasets in this study (see Table 3).

***Flow-based data:*** Flow-level features are extracted in this dataset source. These features are generated by aggregating packets that share common flow identifiers, characterised by the five-tuple: Source IP, Source Port, Destination IP, Destination Port, and Protocol. This five-tuple serves as the flow identifier. The sequence of packets captures the unidirectional or bidirectional flow of communication between two unique endpoints. Common features in this category are statistical features such as bytes per second, bytes per packet, SYN flag count, etc, which tend to provide detailed information about the communication session between two hosts (sender and receiver). As the dominant paradigm (15 of the 22 datasets reviewed, including Bot-IoT, NF-BoT-IoT-v2 and CICIoT2023), flow features balance granularity and scalability by summarising session characteristics such as duration, packet/byte counts, protocol-specific metrics (e.g., DNS query/response ratios), and Time-windowed behavioural profiles.

Comparing both data sources, the flow-based dataset can save on space and processing due to requiring less storage. These can be seen from the packet-level and feature-level data sources available in the IoMT-TrafficData dataset [59]. The flow-based data source required only 10% of the storage needed for the packet-based data.

**Table 3.** Comparison of IoT-focused NIDS datasets

| Dataset name | Year (Released) | IoT as victim or attacker | # Features extracted (labels) | Feature types | Network sniffer | Feature extraction tools (pcap parsing) |
|---|---|---|---|---|---|---|
| N-BaIoT | 2018 | victim | 115 | packet level | Wireshark | custom (adopted in AfterImage) |
| Kitsune | 2018 | victim | 115 | packet level | Tshark | AfterImage |
| BoT-IoT | 2019 | victim | 43 | flow level | Tshark | Argus + inhouse SQL function |
| IoT-SH | 2019 | victim | 121 | packet level | tcpdump | tshark + bespoke Python script |
| IoT-23 | 2020 | victim | 21 | flow level | unknown | Zeek |
| ToN-IoT | 2020 | victim | 44 | flow level | Wireshark | Zeek |
| MQTT-IoT-IDS2020 | 2020 | victim | 31 (bidirectional flow) | 3 levels (packet, unidirectional flow, bidirectional flow) | tcpdump | DPKT (FE source code is public) |
| IoTID20 | 2020 | victim | 83 | flow level | n/a | CICflowmeter |
| MQTTset | 2020 | victim | 33 | packet level | Wireshark | tshark |
| MedBIoT | 2020 | victim | 100 | flow level | tcpdump | proprietary script |
| ECU-IoHT | 2020 | victim | n/a | packet level | Wireshark | n/a |
| TCP FIN Flood and Zbassocflood | 2021 | victim | 95 (wifi) 64 (zigbee) | packet level | Wireshark | Tshark |
| NF-BoT-IoT-v2 and NF-ToN-IoT-v2 | 2021 | n/a | 43 | flow level | n/a | nProbe |
| Edge-IIoTSet | 2022 | victim | 61 | flow level | Wireshark | Zeek and TShark |
| CICIoT2022 | 2022 | victim | 48 | flow level | Wireshark | DPKT |
| AIoT-Sol | 2022 | victim | 83 | flow level | Wireshark | CICflowmeter |
| LATAM-DDoS-IoT | 2022 | victim | 18 | flow level | tcpdump | Argus |
| CICIoT2023 | 2023 | both | 47 | flow level | Wireshark | DPKT |
| GraSec-IOT | 2023 | victim | 83 | flow level | Wireshark | CICFlowmeter |
| IoMT-TrafficData | 2023 | victim | 101 | flow level | tcpdump | Zeek |
| IDSAI | 2023 | victim | 24 | packet level | Wireshark | proprietary script |
| CICIoMT2024 | 2024 | victim | 39 | flow level | Wireshark | DPKT |

**Table 3.** cont.

| Dataset name | # IoT devices in testbed | IoT type | Data formats | Attack scenarios | Average citation/year (Scopus data) | Environment | Reproducible feature extraction? |
|---|---|---|---|---|---|---|---|
| N-BaIoT | 9 | Physical | pcap, csv | Mirai, Bashlite | 147 | Smart Home | no |
| Kitsune | 8 cameras; 9 IoT | physical | pcap, csv | Botnet Malware (Mirai), DoS (SSDP flood, SYN DoS and SSL renegotiation), Reconnaissance (OS scan and Fuzzing), MiTM (Video Injection, ARP MiTM, Active wiretap) | 105 | smart home | yes |
| BoT-IoT | 5 | Simulated | pcap, csv | DDoS, DoS, information gathering, information Theft | 179 | Smart Home | yes |
| IoT-SH | 8 | hysical | - | DoS, MITM/spoofing, reconnaissance, and replay | 69 | Smart Home | yes |
| IoT-23 | 3+ | physical | pcap, log | PartOfAHorizontalPortScan, Mirai, Okiru, DDoS, Torii, C&C, Attack, FileDownload, and HeartBeat | 308 | Smart Home | Yes |
| ToN-IoT | 1(real)<br>5(simulated) | Mixed | csv, pcap | Backdoor, DoS, DDoS, Injection, MITM, Password, Ransomware, Scanning, XSS | 83 | Industrial IoT | Yes |
| MQTT-IoT-IDS2020 | 13 | Simulated | csv, pcap | Aggressive scan, UDP scan, Sparta SSH brute-force, MQTT brute-force attack | 20 | Smart Home | Yes |
| IoTID20 | n/a | n/a | csv | DoS, Mirai, MiTM, and Scan | 37 | Smart Home | Yes |
| MQTTset | 8 | Simulated | pcap, csv | DoS, brute force, malformed data, SlowITe, and flood. | 34 | Smart Home | yes |
| MedBIoT | 80 (Simulated)<br>3 (real) | mixed | pcap, csv | Mirai, BashLite and Torii | 20 | Smart Home | no |
| ECU-IoHT | 3 | physical | xls | Smurf, DoS, ARP spoofing, and Reconnaissance | 9 | Medical IoT | - |
| TCP FIN Flood and Zbassocflood | 6 | physical | pcap | DoS/DDoS | 1 | WSN | yes |
| NF-BoT-IoT-v2 and NF-ToN-IoT-v2 | n/a | n/a | csv | n/a | 37 | n/a | yes |
| Edge-IIoTSet | 10+ | physical | pcap, csv | DDoS/DoS, Reconnaissance, Injection, MiTM, Malware | 123 | Industrial IoT | no |
| CICIoT2022 | 60 | physical | pcap, csv | DoS and RTSP brute-force | 30 | Smart Home | no |
| AIoT-Sol | 6 | physical | csv | DoS, Web attacks, Network Attacks and MQTT attack | 1 | Smart Home | Yes |
| LATAM-DDoS-IoT | 4 (physical),<br>1 (simulated) | Mixed | pcap, csv | DoS, DDoS | 4 | SDN | yes |
| CICIoT2023 | 105 | physical | pcap, csv | DDoS, DoS, BruteForce, Botnet, Infiltration, Web Attacks | 65 | Smart Home | yes |
| GraSec-IOT | 5 | simulated | csv, json | DDoS, Brute force and Port Scanning | 1 | Smart Home | yes |
| IoMT-TrafficData | 4 | physical | pcap, csv | DoS, ARP Spoofing, CAM Table Overflow, MQTT Malaria, Network Scanning, Bluetooth Reconnaissance, and Bluetooth Injection | 0 | Medical IoT | yes |
| IDSAI | unknown | physical | csv | DDoS, brute force, MiTM, UDP port scan, TCP null and DoS | 6 | WSN | no |
| CICIoMT2024 | 39 | Mixed | pcap, csv | DDoS, DoS, Recon, MQTT, and spoofing attacks | 6 | Medical IoT | no |

The distinction between these approaches is critical for NIDS implementation, as packet-level features offer fine-grained detection capabilities at the cost of higher computational overhead, while flow-level features provide scalable analysis suitable for high-speed IoT networks.

### 5.4 *The reproducibility issue in NIDS benchmarking datasets*

Effective deployment of ML-based NIDS hinges on the ability to replicate feature extraction from raw network traffic—a process that must mirror the methodology used in training datasets. While many public datasets, such as IoT-23, ToN-IoT, and NF-BoT-IoT-v2, provide well-documented, tool-based feature extraction (e.g., nProbe, Zeek, etc.), enabling reproducible preprocessing, others suffer from critical limitations.

Datasets like N-BaIoT, Edge-IIoTset, CICIoT2022, CICIoMT2024 and MedBIoT rely on custom, inaccessible scripts, lack sufficient documentation or are overly dependent on environment-specific feature sources, rendering their feature sets irreproducible in different network environments. These constraints render even high-performing NIDS models unusable in practice, as the necessary features for prediction cannot be extracted in a new or varied network environment.

Consequently, datasets with opaque or non-replicable feature extraction processes—including CICIoMT2024 and MedBIoT— should not be employed as benchmarks for real-world applications. Researchers aiming to develop deployable NIDS must prioritise datasets with transparent, tool-driven feature extraction, ensuring their models can transition from experimental validation to operational use. Without addressing these challenges, the disconnect between academic research and practical cybersecurity solution [10] will continue, constraining the effectiveness of ML-based intrusion detection within real-world IoT environments.

## 6. Optimising NIDS solutions: key strategies for effective dataset utilisation and model generalisation

### 6.1 *Provide raw PCAP files*

Though researchers predominantly use CSV-formatted feature sets for ML-based NIDS development, raw PCAP data remains essential, given the lack of standardised feature sets in NIDS research and the diverse feature space extractable from network traffic. The raw PCAPs allow for investigating features with better environmental generalisability. Also, pre-extracted feature sets often contain dataset-specific artifacts [17] and non-reproducible processing pipelines that limit transfer learning. Ref. [10] findings, show that models trained on researcher-derived features outperformed those using authors' feature sets.

### 6.2 *Prioritise cross-dataset-validation*

**Table 4.** Performance metrics from studies that perform cross-dataset validation

| Study | Training dataset | Testing dataset | ML algorithm | Task type | Same-dataset performance | Cross-dataset performance | Performance drop |
|---|---|---|---|---|---|---|---|
| [61] | IDSAI | BoT-IoT | XGBoost | Binary | Accuracy: 94.97%<br>F1: 0.9497 | Accuracy: 94.8%,<br>F1: 0.948 | -0.18% F1 |
| [74] | CICIDS2017 | USB-IDS-1 | RF | Multi-class | F1: 0.99 | F1: 0 - 0.46 | -0.53% to -100% F1 |
| [75] | CICIDS2018 | CICIDS2017 | Autoencoder | Binary | F1: 0.9154 | F1: 0.7705 | -16% F1 |
| [76] | NSL-KDD | guruKDD | RF | Binary | F1: 0.9956 | F1: 0.3608 | -63% F1 |

Dataset creators should implement a rigorous cross-dataset evaluation process wherein models trained on novel datasets are systematically validated against established benchmark datasets. This practice should become standard for all emerging IoT NIDS benchmark datasets, as evidenced by the significant differences in generalisation performance shown in Table 4.

Table 4 provides critical insights into model transferability across different datasets. These findings empirically validate that high performance in the same dataset does not guarantee real-world applicability. The dramatic variance in cross-dataset performance, from near-perfect transfer [61] to complete failure [74]—highlights that cross-dataset validation provides critical insights that single-dataset testing cannot reveal. Without such validation, researchers risk developing models that appear highly effective but fail catastrophically when deployed in different network environments, rendering them unsuitable for practical IoT security implementations.

Cross-dataset evaluation validates detection models and ensures datasets capture sufficiently general attack signatures rather than dataset-specific artifacts [74]. This is particularly crucial for IoT environments where the attack surface evolves rapidly [77], requiring NIDS solutions that generalise effectively to novel threat manifestations.

Our analysis of the 22 IoT-focused NIDS datasets in this study (see Table 3) reveals that most dataset authors did not perform this step when evaluating the efficacy of their dataset for developing NIDS models.

### 6.3 *Documentation is key*

To mitigate the risk of trained NIDS models becoming overly reliant on dataset-specific artifacts that can lead to inflated performance metrics, it is crucial to implement cross-evaluation of models across diverse datasets that reflect different network configurations. To facilitate this, dataset authors must meticulously document key aspects of their process, such as network topologies, attack parameters, feature extraction techniques, and labelling methodologies. This comprehensive documentation is essential for ensuring reproducibility in research.

The IDSAI-BoT-IoT validation paradigm [61] demonstrates how a transparent methodology can foster robust cross-dataset evaluations and enhance model development. In IoT security research, where device heterogeneity and the dynamic nature of threats demand stringent validation processes, such detailed documentation should be regarded as a fundamental necessity rather than an optional addendum.

### 6.4 *Need for a standardised feature*

There is a need for standardised features in benchmarking datasets. The current reliance on ad hoc, expert-defined feature sets—extracted through manual trial-and-error processes [9]—introduces critical limitations in evaluating ML-based NIDS. Without standardised features, comparative benchmarking across datasets becomes unreliable, as divergent feature spaces obscure true model generalizability [10]. This fragmentation perpetuates a research-production gap, where models optimised for dataset-specific artefacts fail in real-world deployments.

A universal feature standard would address three core challenges: (1) eliminating redundant or irrelevant features that lack security relevance, (2) enabling cross-dataset validation of model performance, and (3) facilitating the creation of multi-environment datasets that reflect operational networks. As demonstrated by [10], such standardisation is a prerequisite for translating academic research into deployable NIDS solutions. This remains an area of open research.

## 7. Conclusion

This study provides a comprehensive review of benchmarking datasets used in IoT security, with a focus on their testbed configurations and feature extraction methodologies. The fluctuating publication trends of IoT NIDS datasets, as illustrated in Figure 2, highlight the irregular pace of new dataset development, peaking in 2020 without consistent growth. Through critical analysis, the study identified significant gaps between academic research and operational deployment, including notable inconsistencies in feature selection among datasets using the same tools (e.g., Argus, CICFlowMeter), non-reproducible extraction methods in approximately 27% of datasets, and persistent discrepancies in feature naming conventions. These findings underscore the urgent need for standardised feature sets to support fair evaluation and facilitate cross-study comparisons. In response, this study recommends practical measures to enhance dataset design and documentation, aiming to shift NIDS development from an accuracy-focused approach to one prioritising reproducibility—thereby accelerating the translation of academic advancements into effective, real-world cybersecurity solutions.

## Confilict of interest

The authors declare no competing financial interest.

## References

[1] F. Schuster and A. Habibipour, "Users' privacy and security concerns that affect iot adoption in the home domain," *International Journal of Human-Computer Interaction*, vol. 40, no. 7, pp. 1632-1643, 2024.

[2] L. S. Vailshery, "Number of internet of things (iot) connections worldwide from 2022 to 2023, with forecasts from 2024 to 2033," 2024. [Online]. Available: https://www.statista.com/statistics/1183457/iot-connected-devices-worldwide/. [Accessed Oct. 7, 2024].

[3] K. S. Mohamed, *The Era of Internet of Things: Towards a Smart World*. Cham: Springer International Publishing, 2019, pp. 1-19.

[4] P. Paganini, "Botnets and cybercrime - introduction," 2013. [Online]. Available: https://www.infosecinstitute.com/resources/general-security/botnets-and-cybercrime-introduction/. [Accessed Apr. 22, 2024].

[5] L. H. Newman, "What we know about friday's massive east coast internet outage," 2016. [Online]. Available: https://www.wired.com/2016/10/internet-outage-ddos-dns-dyn/. [Accessed Apr. 22, 2024].

[6] A. Wajeeha, "Why botnets persist: designing effective technical and policy interventions," 2019. [Online]. Available: https://internetpolicy.mit.edu/why-botnets-persist-designing-effective-technical-and-policy-interventions/. [Accessed Jan. 14, 2025].

[7] P. M. S. Sánchez, J. M. J. Valero, A. H. Celdrán, G. Bovet, M. G. Pérez, and G. M. Pérez, "A survey on device behavior fingerprinting: data sources, techniques, application scenarios, and datasets," *IEEE Communications Surveys & Tutorials*, vol. 23, no. 2, pp. 1048-1077, 2021.

[8] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, A. Shabtai, D. Breitenbacher, et al., "N-baiot—network-based detection of iot botnet attacks using deep autoencoders," *IEEE Pervasive Computing*, vol. 17, no. 3, pp. 12-22, 2018.

[9] Y. N. Kunang, S. Nurmaini, D. Stiawan, and Y. B. Suprapto, "Improving classification attacks in iot intrusion detection system using bayesian hyperparameter optimization," in Proc. 3rd International Seminar on Research of Information Technology and Intelligent Systems, Yogyakarta, Indonesia, Dec. 2020.

[10] M. Sarhan, S. Layeghy, and M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Networks and Applications*, vol. 27, no. 1, pp. 1-14, 2022.

[11] Y. Mirsky, T. Doitshman, Y. Elovici, and A. Shabtai, "Kitsune: an ensemble of autoencoders for online network intrusion detection," *arXiv preprint arXiv:1802.09089*, 2018.

[12] A. Binbusayyis and T. Vaiyapuri, "Identifying and benchmarking key features for cyber intrusion detection: an ensemble approach," *IEEE Access*, vol. 7, pp. 106495-106513, 2019.

[13] A. Khacha, Z. Aliouat, Y. Harbi, C. Gherbi, R. Saadouni, and S. Harous, "Landscape of learning techniques for intrusion detection system in iot: a systematic literature review," *Computers and Electrical Engineering*, vol. 120, p. 109725, 2024.

[14] A. Khraisat and A. Alazab, "A critical review of intrusion detection systems in the internet of things: techniques, deployment strategy, validation strategy, attacks, public datasets and challenges," *Cybersecurity*, vol. 4, no. 1, pp. 1-27, 2021.

[15] A. L. Buczak and E. Guven, "A survey of data mining and machine learning methods for cyber security intrusion detection," *IEEE Communications Surveys & Tutorials*, vol. 18, no. 2, pp. 1153-1176, 2015.

[16] N. Chaabouni, M. Mosbah, A. Zemmari, C. Sauvignac, and P. Faruki, “Network intrusion detection for iot security based on learning techniques,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2671-2701, 2019.

[17] M. Ring, S. Wunderlich, D. Scheuring, D. Landes, and A. Hotho, “A survey of network-based intrusion detection data sets,” *Computer Security*, vol. 86, pp. 147-167, 2019.

[18] S. Haque, F. El-Moussa, N. Komninos, and R. Muttukrishnan, “A systematic review of data-driven attack detection trends in iot,” *Sensors*, vol. 23, no. 16, p. 7191, 2023.

[19] M. J. Page, J. E. McKenzie, P. M. Bossuyt, I. Boutron, T. C. Hoffmann, C. D. Mulrow, et al., “The PRISMA 2020 statement: an updated guideline for reporting systematic reviews,” *International Journal of Surgery*, vol. 88, p. 105906, 2021.

[20] C. Wohlin, “Guidelines for snowballing in systematic literature studies and a replication in software engineering,” in Proc. 18th International Conference on Evaluation and Assessment in Software Engineering, London, England, May 2014.

[21] E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, and A. A. Ghorbani, “CICIoT2023: a real-time dataset and benchmark for large-scale attacks in iot environment,” *Sensors*, vol. 23, no. 13, p. 5941, 2023.

[22] Y. Meidan, M. Bohadana, Y. Mathov, Y. Mirsky, D. Breitenbacher, and A. Shabtai, “Detection of iot botnet attacks (N-BaIoT),” *UCI Machine Learning Repository*, 2018.

[23] Y. Mirsky, “AfterImage,” 2020. [Online]. Available: https://github.com/ymirsky/Kitsune-py/blob/master/AfterImage.py. [Accessed Mar. 6, 2025].

[24] Y. Mirsky, “Kitsune network attack dataset,” 2020. [Online]. Available: https://www.kaggle.com/datasets/ymirsky/network-attack-dataset-kitsune. [Accessed Mar. 6, 2025].

[25] N. Koroniotis, N. Moustafa, E. Sitnikova, and B. Turnbull, “Towards the development of realistic botnet dataset in the internet of things for network forensic analytics: bot-iot dataset,” *Future Generation Computer Systems*, vol. 100, pp. 779-796, 2019.

[26] N. Moustafa, “The bot-iot dataset,” *IEEE Dataport*, 2019.

[27] E. Anthi, L. Williams, M. Słowińska, G. Theodorakopoulos, and P. Burnap, “A supervised intrusion detection system for smart home iot devices,” *IEEE Internet of Things Journal*, vol. 6, no. 5, pp. 9042-9053, 2019.

[28] irini90, “pcap_preprocessing,” 2018. [Online]. Available: https://github.com/irini90/pcap_preprocessing/blob/master/pdml2arff.py. [Accessed Mar. 6, 2025].

[29] S. Garcia, A. Parmisano, and M. J. Erquiaga, “IoT-23: a labeled dataset with malicious and benign iot network traffic,” 2020.

[30] Stratosphere Lab, “IoT-23 dataset: a labeled dataset of malware and benign iot traffic,” 2020. [Online]. Available: https://www.stratosphereips.org/datasets-iot23. [Accessed Mar. 26, 2025].

[31] I. Ullah and Q. H. Mahmoud, “A scheme for generating a dataset for anomalous activity detection in iot networks,” in *Canadian Conference on Artificial Intelligence*. Springer; 2020.

[32] H. Kang, D. H. Ahn, G. M. Lee, J. D. Yoo, K. H. Park, and H. K. Kim, “Iot network intrusion dataset,” 2019. [Online]. Available: https://ieee-dataport.org/open-access/iot-network-intrusion-dataset. [Accessed Mar. 6, 2024].

[33] A. H. Lashkari, G. D. Gil, M. S. I. Mamun, and A. A. Ghorbani, “Characterization of tor traffic using time based features,” in *International Conference on Information Systems Security and Privacy*. SciTePress; 2017

[34] I. Ullah and Q. Mahmoud, “Iot intrusion dataset,” 2020. [Online]. Available: https://sites.google.com/view/iot-network-intrusion-dataset/home. [Accessed Mar. 6, 2025].

[35] N. Moustafa, “A new distributed architecture for evaluating ai-based security systems at the edge: network TON_IoT datasets,” *Sustainable Cities and Society*, vol. 72, p. 102994, 2021.

[36] T. M. Booij, I. Chiscop, E. Meeuwissen, N. Moustafa, and F. T. Den Hartog, “ToN_IoT: the role of heterogeneity and the need for standardization of features and attack types in iot network intrusion data sets,” *IEEE Internet of Things Journal*, vol. 9, no. 1, pp. 485-496, 2021.

[37] A. Alsaedi, N. Moustafa, Z. Tari, A. Mahmood, and A. Anwar, “TON_IoT telemetry dataset: a new generation dataset of iot and iiot for data-driven intrusion detection systems,” *IEEE Access*, vol. 8, pp. 165130-165150, 2020.

[38] N. Moustafa, “The TON_IoT datasets,” 2021. [Online]. Available: https://research.unsw.edu.au/projects/toniot-datasets. [Accessed Mar. 27, 2025].

[39] I. Vaccari, G. Chiola, M. Aiello, M. Mongelli, and E. Cambiaso, “MQTTset, a new dataset for machine learning techniques on MQTT,” *Sensors*, vol. 20, no. 22, p. 6578, 2020.

[40] CNR-IEIIT, "MQTTset," 2020. [Online]. Available: https://www.kaggle.com/datasets/cnrieiit/mqttset. [Accessed Apr. 8, 2025].

[41] H. Hindy, E. Bayne, M. Bures, R. Atkinson, C. Tachtatzis, and X. Bellekens, "Machine learning based iot intrusion detection system: an MQTT case study (MQTT-IoT-IDS2020 dataset)," in *International Networking Conference*. Springer; 2020.

[42] Abertay Machine Learning Group, "Machine learning based iot intrusion detection system: an MQTT case study," 2021. [Online]. Available: https://github.com/AbertayMachineLearningGroup/MQTT_ML. [Accessed Apr. 8, 2025].

[43] H. Hanan, C. Tachtatzis, R. Atkinson, E. Bayne, et al., "MQTT-IoT-IDS2020: MQTT internet of things intrusion detection dataset," 2020. [Online]. Available: https://dx.doi.org/10.21227/bhxy-ep04. [Accessed Apr. 8, 2025].

[44] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "MedBIoT: generation of an iot botnet dataset in a medium-sized iot network," in *Information Systems Security and Privacy*. SciTePress; 2020.

[45] A. Guerra-Manzanares, J. Medina-Galindo, H. Bahsi, and S. Nõmm, "MedBIoT: generation of an iot botnet dataset in a medium-sized iot network," 2020. [Online]. Available: https://cs.taltech.ee/research/data/medbiot. [Accessed Apr. 1, 2025].

[46] M. Ahmed, S. Byreddy, A. Nutakki, L. F. Sikos, and P. Haskell-Dowland, "ECU-IoHT: a dataset for analyzing cyberattacks in internet of health things," *Ad Hoc Networks*, vol. 122, p. 102621, 2021.

[47] M. Ahmed, S. Byreddy, A. Nutakki, L. Sikos, and P. Haskell-Dowland, "ECU-IoHT," 2020. [Online]. Available: https://ro.ecu.edu.au/datasets/48. [Accessed Mar. 31, 2025].

[48] D. Stiawan, D. Wahyudi, T. W. Septian, M. Y. Idris, and R. Budiarto, "The development of an internet of things (iot) network traffic dataset with simulated attack data," *Journal of Internet Technology*, vol. 24, no. 2, pp. 345-356, 2023.

[49] D. Stiawan, "TCP FIN flood and Zbassocflood dataset," 2021. [Online]. Available: https://ieee-dataport.org/documents/tcp-fin-flood-and-zbassocflood-dataset. [Accessed Mar. 29, 2025].

[50] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, H. Janicke, "Edge-IIoTset: a new comprehensive realistic cyber security dataset of iot and iiot applications for centralized and federated learning," *IEEE Access*, vol. 10, pp. 40281-40306, 2022.

[51] S. Dadkhah, H. Mahdikhani, P. K. Danso, A. Zohourian, K. A. Truong, A. A. Ghorbani, et al., "Towards the development of a realistic multidimensional IoT profiling dataset," in *19th Annual International Conference on Privacy, Security & Trust (PST)*. Fredericton, Canada: IEEE; 2022.

[52] Canadian Institute for Cybersecurity, "CIC IoT Dataset 2022," 2022. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2022.html. [Accessed Mar. 28, 2025].

[53] University of Queensland, "ML-based NIDS datasets," 2025. [Online]. Available: https://www.itee.uq.edu.au/research/cyber-security/research-areas. [Accessed Apr. 2, 2025].

[54] N. M. Min, V. Visoottiviseth, S. Teerakanok, and N. Yamai, "OWASP IoT Top 10 based attack dataset for machine learning," In Proc. 24th International Conference on Advanced Communication Technology (ICACT), PyeongChang, Korea, Feb. 13-16, 2022.

[55] J. G. Almaraz-Rivera, J. A. Perez-Diaz, J. A. Cantoral-Ceballos, J. F. Botero, and L. A. Trejo, "Toward the protection of IoT networks: introducing the LATAM-DDoS-IoT dataset," *IEEE Access*, vol. 10, pp. 106909-106920, 2022.

[56] J. G. Almaraz-Rivera, J. A. Perez-Diaz, and J. A. Cantoral-Ceballos, "Transport and application layer DDoS attacks detection to IoT devices by using machine learning and deep learning models," *Sensors*, vol. 22, no. 9, p. 3367, 2022.

[57] J. G. Almaraz-Rivera, J. A. Perez-Diaz, J. A. Cantoral-Ceballos, J. F. Botero, and L. A. Trejo, "LATAM-DDoS-IoT dataset," 2022. [Online]. Available: https://ieee-dataport.org/documents/latam-ddos-iot-dataset. [Accessed Mar. 30, 2025].

[58] Canadian Institute for Cybersecurity, "CIC IoT Dataset 2023," 2023. [Online]. Available: https://www.unb.ca/cic/datasets/iotdataset-2023.html. [Accessed Mar. 28, 2025].

[59] J. Areia, I. Bispo, L. Santos, and R. L. da Costa, "IoMT-TrafficData: dataset and tools for benchmarking intrusion detection in internet of medical things," *IEEE Access*, vol. 12, pp. 115370-115385, 2024.

[60] J. Areia, I. A. Bispo, L. Santos, and R. L. Costa, "IoMT-TrafficData: a dataset for benchmarking intrusion detection in IoMT," *Zenodo*, 2023, https://doi.org/10.5281/zenodo.8116338.

[61] G. Fernando, A. H. Brayan, M. Florina, C. Liliana, G.-A. Héctor, and T. Reinel, "Enhancing intrusion detection in IoT communications through ML model generalization with a new dataset (IDSAI)," *IEEE Access*, vol. 11, pp. 123456-123467, 2023.

[62] BioAITeam, "Intrusion-detection-system-using-machine-learning," 2023. [Online]. Available: https://github.com/BioAITeam/Intrusion-Detection-System-using-Machine-Learning/commits/main/. [Accessed Mar. 29, 2025].

[63] D. Hamouche, R. Kadri, M. Messai, and H. Seba, "(POSTER) A graph dataset for security enforcement in IoT networks: GRASEC-IoT," In Proc. 20th International Conference on Distributed Computing in Smart Systems and the Internet of Things (DCOSS-IoT), Washington DC, USA, Apr. 29-May 1, 2024.

[64] Gladis, "GraSec-IOT," 2024. [Online]. Available: https://gitlab.liris.cnrs.fr/gladis/grasec-iot. [Accessed Jan. 11, 2025].

[65] S. Dadkhah, E. C. P. Neto, R. Ferreira, R. C. Molokwu, S. Sadeghi, and A. A. Ghorbani, "CICIoMT2024: a benchmark dataset for multi-protocol security assessment in IoMT," *Internet of Things*, vol. 28, p. 101351, 2024.

[66] Canadian Institute for Cybersecurity, "CIC IoMT Dataset 2024," 2024. [Online]. Available: https://www.unb.ca/cic/datasets/iomt-dataset-2024.html. [Accessed Mar. 28, 2025].

[67] hieulw, "cicflowmeter," 2024. [Online]. Available: https://github.com/hieulw/cicflowmeter. [Accessed May 1, 2025].

[68] Ahlashkari, "CICFlowMeter," 2022. [Online]. Available: https://github.com/ahlashkari/CICFlowMeter. [Accessed May 1, 2025].

[69] Zeek, "The Zeek network security monitor," 2025. [Online]. Available: https://zeek.org/. [Accessed Apr. 8, 2025].

[70] ntop, "Command line options—nProbe 10.1 documentation," 2025. [Online]. Available: https://www.ntop.org/guides/nprobe/cli_options.html. [Accessed Apr. 8, 2025].

[71] ntop, "nProbe," 2025. [Online]. Available: https://github.com/ntop/nProbe. [Accessed Apr. 8, 2025].

[72] C. Bullard, "Argus," 2024. [Online]. Available: https://github.com/openargus/argus. [Accessed Apr. 8, 2025].

[73] K. Bandla, "DPKT," 2024. [Online]. Available: https://github.com/kbandla/dpkt. [Accessed Apr. 8, 2025].

[74] M. Catillo, A. Del Vecchio, A. Pecchia, and U. Villano, "Transferability of machine learning models learned from public intrusion detection datasets: the CICIDS2017 case study," *Software Quality Journal*, vol. 30, no. 4, pp. 955-981, 2022.

[75] M. Verkerken, L. D' hooge, T. Wauters, B. Volckaert, and F. De Turck, "Towards model generalization for intrusion detection: unsupervised machine learning techniques," *Journal of Network and Systems Management*, vol. 30, no. 1, p. 12, 2021.

[76] S. Al-Riyami, A. Lisitsa, and F. Coenen, "Cross-datasets evaluation of machine learning models for intrusion detection systems," in *Proceedings of Sixth International Congress on Information and Communication Technology*. London, UK: Springer; 2022.

[77] A. Bhardwaj, K. Kaushik, V. Dagar, and M. Kumar, "Framework to measure and reduce the threat surface area for smart home devices," *Advances in Computational Intelligence*, vol. 3, no. 4, p. 16, 2023.