Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Dataset Quality in Machine Learning for Intrusion Detection: An Empirical Benchmarking Study, Design Guidelines, and a Testbed-Validated Pilot

Ifrah Sanober

ifrah_2022phacse007@nitsri.ac.in

National Institute of Technology Srinagar

Roohie Naaz Mir

National Institute of Technology Srinagar

Research Article

Keywords: data quality, machine learning datasets, class imbalance, data leakage, benchmark evaluation, intrusion detection

Posted Date: November 20th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-10642513/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Dataset Quality in Machine Learning for Intrusion Detection: An Empirical Benchmarking Study, Design Guidelines, and a Testbed-Validated Pilot

Ifrah Sanober[1*] and Roohie Naaz Mir[1]

[1*]Department of Computer Science and Engineering, National Institute of Technology Srinagar, Srinagar, Jammu & Kashmir, India.

*Corresponding author(s). E-mail(s): ifrah_2022phacse007@nitsri.ac.in;
Contributing authors: naaz310@nitsri.ac.in;

Abstract

The reliability of machine learning models depends fundamentally on the quality of the datasets used to train and evaluate them, yet dataset quality is often assumed rather than measured. This problem is acute in machine-learning-based intrusion detection, where widely used benchmark datasets suffer from severe class imbalance, inconsistent labelling, data leakage and limited representativeness, pathologies that systematically distort model evaluation. Through a systematic benchmarking study of five datasets across four machine learning and deep learning models, this paper quantifies how these data-quality deficits undermine evaluation: aggregate accuracy diverges from macro-averaged F1 by as much as 0.63, and dataset choice drives more performance variance than model architecture. Building on these findings, this study proposes six quantitative, verifiable design guidelines for constructing high-quality datasets, covering class balance, encrypted-traffic representation, adversarial coverage, multimodal observability, annotation quality and realistic data dynamics. It also operationalises them through a nine-stage data-construction pipeline. The guidelines are validated on a pilot dataset built from a ten-node virtualised testbed, incorporating a per-class deduplication step that exposes how raw sample counts overstate effective data diversity, a data-leakage audit and multi-seed evaluation. The pilot exhibits structural properties: encrypted traffic, eight adversary tactics,

and a complete multi-stage attack chain, which are absent from existing benchmarks, demonstrating that the proposed data-quality principles are achievable in practice.

Keywords: data quality, machine learning datasets, class imbalance, data leakage, benchmark evaluation, intrusion detection

1 Introduction

The performance of any machine learning (ML) or deep learning (DL) system is bounded by the quality of the data on which it is trained and evaluated. Where datasets are imbalanced, mislabelled, or unrepresentative of the phenomena they purport to capture, reported model performance can diverge sharply from real-world behaviour: a concern spans the whole of data-driven research. This dependence is especially consequential in security analytics, where ML-based intrusion detection has become a dominant paradigm and where the Internet of Things (IoT) has rapidly expanded the volume and heterogeneity of network data requiring analysis [1, 2]. IoT devices are resource-constrained, poorly patched and highly interconnected [3, 4], generating large-scale, high-velocity traffic whose analysis places the underlying dataset quality directly on the critical path to reliable detection [5].

Intrusion detection systems (IDS) have long been integral to cyber defence [6], and modern IDS increasingly rely on ML and DL for attack recognition [7]. Their effectiveness, however, is governed by the datasets used for training and testing: models learned from outdated, skewed, or synthetic data may report strong benchmark performance yet fail in deployment [8]. Intrusion detection thus offers an especially clear and measurable lens on a general problem in data science. It concerns the gap between apparent and genuine model quality that arises from data deficiencies. This gap is particularly visible because intrusion detection datasets are large, heavily imbalanced, and accompanied by ground-truth labels whose limitations can be examined directly.

Benchmark datasets such as NSL-KDD [9], UNSW-NB15 [10], CIC-IDS2017 [11], CSE-CIC-IDS2018 [12] and HIKARI-2021 [13] have been widely used to advance IDS research. However, each suffers from limitations that limit its relevance to IoT contexts [14]. Moreover, emerging IoT-specific datasets such as BoT-IoT [15] and ToN-IoT [16], while promising, remain narrow in scope and continue to face problems of imbalance and representational bias [17].

This paper makes three contributions.

1. It empirically quantifies how dataset-quality deficits, class imbalance, labelling inconsistency, data leakage and limited representativeness, distort the evaluation of ML and DL models, using intrusion detection as a data-rich, high-stakes case study across five widely used datasets and four models.
2. It proposes six quantitative, verifiable design guidelines for constructing high-quality datasets, translating qualitative data-quality desiderata into checkable acceptance criteria spanning class balance, encrypted-traffic representation, adversarial coverage, multimodal observability, annotation quality, and realistic data dynamics.
3. It operationalises these guidelines as a reproducible data-construction pipeline and validates them on a pilot dataset built from a virtualised testbed, incorporating a per-class deduplication analysis, a data-leakage audit, and multi-seed evaluation to demonstrate that the proposed principles are achievable in practice.

The intention is not to replicate existing evaluation frameworks, but to provide practical, actionable, and generalisable directions for building datasets on which data-driven detection systems can be reliably trained and assessed.

The rest of the paper is organised as follows: Section 2 gives an overview of the related work. Section 3 presents a data-quality analysis of the benchmark datasets. Design guidelines and the roadmap for testbed and dataset generation are given in

Sections 4 and 5, respectively. Section 6 presents the pilot dataset validation. Finally, Section 7 presents the conclusion and future work.

2 Related Work

Research on intrusion detection datasets has largely evolved along two threads: (i) general-purpose network IDS corpora (e.g., NSL-KDD [9], UNSW-NB15 [10], CIC-IDS2017 [11], etc.) and (ii) methodology work that proposes criteria or frameworks to evaluate such corpora. Both threads have moved the field forward, yet important gaps remain, especially for IoT-centric evaluation and dataset design.

2.1 Benchmarks

Despite the widespread use of classic datasets in benchmarking efforts, several papers point to inherent problems [18]. While NSL-KDD is widely employed historically, it shows signs of being outdated with regards to traffic realism [19], UNSW-NB15 provides more contemporary attacks, yet lacks balanced features and protocol representations [20]; CIC-IDS2017 offers rich features, but does not provide realistic backgrounds [21]; finally, HIKARI-2021, while recently collected, suffers from insufficiently diverse attacks [22]. The empirical analyses prove such a claim by showing that feature and protocol completeness vary significantly across different corpora and volume realism is rather poor, hence questioning external validity of the research results.

2.2 Frameworks and Dataset Evaluation

Methodology papers have proposed qualitative checklists, covering protocol availability, capture completeness, metadata, labelling practice, and attack diversity, to guide dataset generation and assessment [23, 24] (influencing, for instance, how CIC-IDS2017 and related corpora were designed); the present work builds directly on the authors'

own earlier conceptual framework [24], which it extends from a qualitative checklist into the quantitative, empirically-grounded guidelines developed here. These efforts help to articulate what 'good' should look like, but they are largely descriptive and nonquantitative, offering limited means to score datasets on standardised and reproducible scales or to address specific constraints of IoT (constrained devices, specialised protocols, edge-to-cloud patterns).

2.3 IoT-focused Needs

Relative to enterprise NIDS corpora, IoT deployments introduce device heterogeneity, resource constraints, and protocol variation (e.g., lightweight application protocols and short-burst telemetry) [25]. Cross-dataset comparisons show that existing benchmarks rarely highlight salient properties of the IoT. In particular, there is little explicit coverage of: (i) IoT scenarios and device behaviours, (ii) zero-day/APT-like multi-stage sequences, (iii) adversarial/evasion attempts, and (iv) multimodal telemetry (packet/flow + host/app logs + device states) [16]. Even where encryption appears, it is often partial, and user-behaviour signals are largely absent.

Beyond coverage, operational realism is underserved for the IoT [26]. Benchmarks rarely include: (a) encrypted traffic analytics appropriate for constrained devices; (b) privacy-preserving release mechanisms; (c) mechanisms to keep attack labels and TTP mappings current with evolving threat intelligence; and (d) ground-truth practices robust to noisy labels and heterogeneous networks. These omissions impede the credible generalisation of IDS models from laboratory settings to production IoT [17, 27].

More recent IoT-specific corpora have sought to address several of these gaps. Edge-IIoTset [28] provides a large-scale dataset spanning multiple IoT and IIoT protocols with a layered feature set; CICIoT2023 [29] captures a wide range of attacks

executed against a topology of real IoT devices; and IoT-23 [30] offers labelled malware-capture traffic from consumer IoT devices. These datasets represent clear progress in scale and device realism. However, they continue to exhibit the structural limitations documented above: encrypted-traffic coverage remains partial, multimodal host- and application-layer telemetry is largely absent, class distributions remain severely imbalanced, and labelling is generally single-dimensional rather than aligned to a structured tactic taxonomy. Their persistence despite the availability of newer corpora indicates that the gaps this paper addresses are not merely a function of dataset age, but reflect the absence of an explicit, quantitative design framework guiding dataset construction, which is precisely the contribution developed in Sections 4–6.

2.4 Cross-cutting Issues

Three cross-cutting problems recur across the literature and become acute in IoT contexts:

- **Label and taxonomy alignment.** Attack labels are inconsistent across datasets and rarely cleanly mapped to a common taxonomy (e.g., ATT&CK) [31], complicating transfer learning and comparative evaluation and hindering the measurement of attack-breadth coverage over IoT-specific TTPs.
- **Feature schema fragmentation.** Feature sets differ substantially (often lacking host/app layer or device-behaviour fields), forcing ad hoc mappings and restricting the detection of stealthy or protocol-specific IoT attacks [32, 33].
- **Traffic distribution mismatch.** Synthetic or staged traffic typically diverges from real deployments, degrading the external validity of benchmarking results, especially for IoT burstiness, short flows and edge-to-cloud patterns. Dataset audits repeatedly expose this realism gap [34].

2.5 Gap Summary

Existing literature defines useful criteria and some metrics, but IoT-centric datasets remain underspecified on scenario coverage, signals/modalities, privacy-preserving sharing, currency (threat-intel alignment) and realism/balance. This paper responds by (i) examining widely used corpora through an IoT-aware lens and (ii) distilling concrete, checkable design guidelines plus a testbed roadmap for building realistic, balanced, and multimodal IoT IDS datasets.

3 IoT-Aware Analysis of Popular IDS Benchmarks

To ground the limitations of widely used corpora in evidence rather than assertion, a systematic benchmarking study was conducted across five datasets — NSL-KDD [35], UNSW-NB15 [36], CIC-IDS2017 [37], BoT-IoT [38], and ToN-IoT [39] — using four representative models: Random Forest (RF), XGBoost (XGB), a one-dimensional convolutional neural network (1D-CNN), and a long short-term memory network (LSTM). All experiments were conducted on fixed 80/20 stratified splits, with class-balanced training throughout. Per-class precision, recall, and F1-score were the primary evaluation metrics; accuracy and weighted-F1 are reported alongside for completeness but are treated with caution given the severe imbalance across all five corpora.

The findings are organised below along four dimensions: structural profile, class imbalance, adversary-tactic coverage, and the sensitivity of measured performance to dataset choice. Table 1 summarises the model configurations used throughout. Tree-based models used scikit-learn defaults with balanced class weighting; the deep models used the architectures and training schedule shown, with class-weighted cross-entropy loss, early stopping (patience 5), and learning-rate reduction on plateau. All experiments used a fixed random seed for reproducibility, and the complete configuration is available in the accompanying repository.

Table 1 Model Configurations Used in the Benchmarking Study

Model	Configuration
Random Forest	200 trees, unbounded depth, balanced class weight
XGBoost	300 trees, max depth 6, learning rate 0.1, multi:softmax, sample-weighted
1D-CNN	2 conv blocks (64, 128 filters, kernel 3) + global pooling + dense(128); Adam, lr 0.001; batch 256; up to 50 epochs
LSTM	2 LSTM layers (128, 64 units) + dense(64); Adam, lr 0.001; batch 256; up to 50 epochs

3.1 Structural Profile

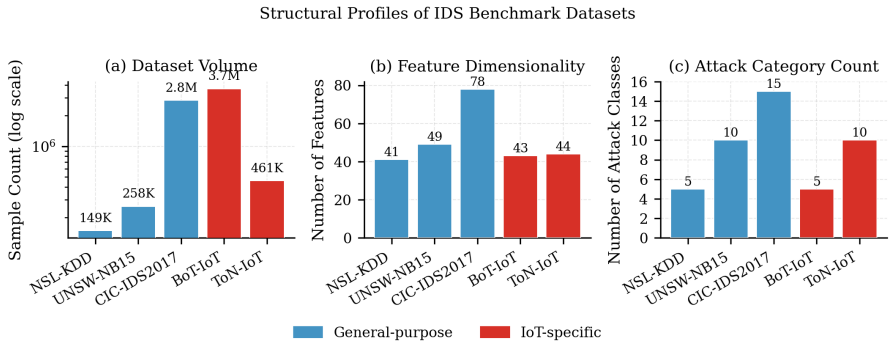


Fig. 1 Structural profiles of the five benchmark datasets, including sample volume, feature count, and class count.

Figure 1 demonstrates differences between structural characteristics of each of the datasets. There are two important findings with immediate implications for applicability to IoT scenarios. First of all, the feature schema differs considerably among datasets. The NSL-KDD dataset includes 41 features derived from traffic captured for the KDD Cup 1999 dataset and reflecting a network environment which is almost twenty years old compared to today's IoT deployments. UNSW-NB15 and CIC-IDS2017 rely on more comprehensive flow-level schemas but still stay tied to the protocol schemes which were traditionally used within enterprises. The BoT-IoT and ToN-IoT, although labeled as IoT datasets, utilize IP flow-level schema without having significant representation of application-layer protocols such as MQTT, CoAP, and

AMQP used by constrained devices. Secondly, the number and degree of separation of attack categories range from five in NSL-KDD to fifteen in CIC-IDS2017.

3.2 Class Imbalance and Rare-Event Blind Spots (Diagnosis)

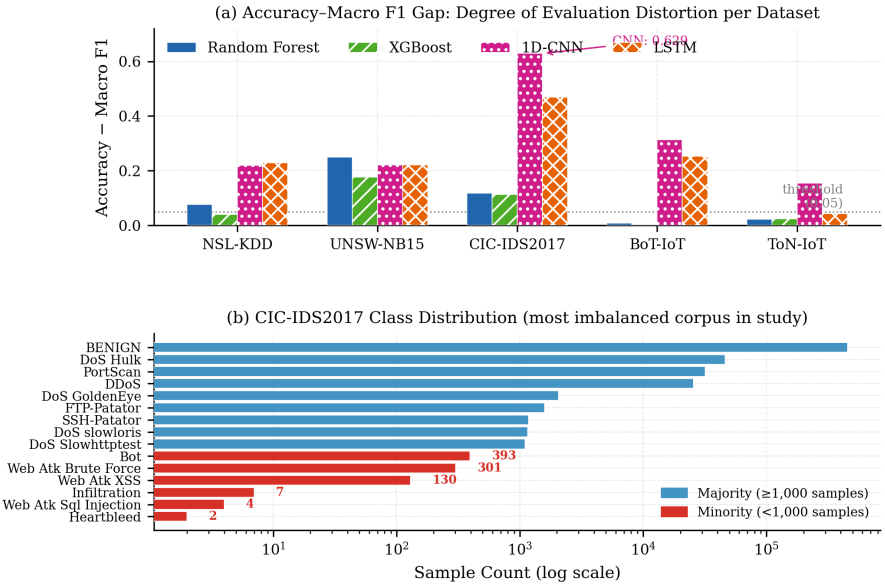


Fig. 2 Class distribution across benchmark datasets (log scale). Imbalance ratios range from moderate to extreme; the accuracy—macro-F1 gap quantifies the resulting evaluation distortion.

Across all five datasets, class distributions are heavily skewed, with a small number of majority-class flows accounting for the bulk of samples, while several operationally critical categories are represented by a handful of instances. Figure 2 illustrates this pattern. The practical consequences are documented directly in the experimental results: every dataset in this study produced an accuracy—macro-F1 gap exceeding 0.05 when averaged across models, with CIC-IDS2017 recording the largest discrepancy at 0.3324 on average. For the 1D-CNN specifically, overall accuracy on CIC-IDS2017 was 0.7579 while macro-F1 was 0.1288, a gap of 0.6291, demonstrating that a model can appear to perform well under aggregate measurement while failing on most of the attack categories it is supposed to detect.

The sources of this gap are identifiable at the class level. In CIC-IDS2017, several categories such as Bot (393 samples), Infiltration (7 samples), Web Attack-Sql Injection (4 samples), and Heartbleed (2 samples) are represented so sparsely relative to the BENIGN class (454,620 samples) that even well-tuned models with balanced training exhibit near-zero recall for these classes. Heartbleed, with 2 test-set samples, is borderline unevaluable under standard split conditions. Similarly, in BoT-IoT, the Theft category contains 16 test samples against 385,325 DDoS flows; no model achieved an F1-score above 0.12 on Theft despite accuracy approaching unity. In NSL-KDD, the U2R class (22 test samples) produced F1-scores of 0.027 (LSTM) and 0.071 (CNN) — two of the worst single-class outcomes across the entire study.

Three methodological implications follow. First, accuracy should not appear as a primary metric in IDS evaluations involving imbalanced data; macro-F1 or per-class reporting is required for any honest assessment. Second, minority classes representing targeted attack types with low sample counts should be explicitly flagged in dataset documentation, as their contribution to macro-averaged scores is disproportionate to their volume. Third, balancing procedures applied without preserving temporal and flow-level structure, as is common in the literature, mask rather than resolve the underlying representational problem.

3.3 Coverage of Adversary Tactics

Per-class F1 scores, depicted in Figure 3, reveal clear gaps in adversary-tactic coverage. The four models exhibit poor performance on particular classes, namely U2R attacks in NSL-KDD (F1 = 0.667 for the best-performing RF model), Analysis and Worms attacks in UNSW-NB15 (F1 = 0.073 and 0.478 respectively by RF), and Infiltration attacks in CIC-IDS2017 (F1 = 0.923 by RF, but with only 7 instances of test data). These particular classes coincide exactly with the less represented ATT&CK tactic categories of privilege escalation, exploitation-based initial access, and insider



Fig. 3 Per-class F1-score heatmaps across all model–dataset combinations. Classes corresponding to low-frequency or sophisticated attack tactics (e.g., U2R, Analysis, Worms, Infiltration) show consistently poor detection across all models, revealing systematic adversary-tactic coverage gaps in current benchmark corpora.

or low-and-slow infiltration attacks. Attacks that are highly frequent and have unique patterns, such as DDoS, DoS, and Generic, tend to get close to perfection on F1 metric on all models and datasets, overshadowing the other attacks due to their dominance of training signal.

Scenarios involving misuse of IoT protocols are not present in any of the five datasets considered. None of them includes attacks related to command injection via MQTT, amplification using CoAP, or firmware attacks as part of the supply chain attacks, which are commonly found in actual IoT breaches. Despite being designed specifically for IoT systems, BoT-IoT and ToN-IoT restrict their coverage to volumetric attacks (DDoS and DoS) and attacks on credentials (brute force and scanning). This limitation is particularly important for performance in detection because models

trained and tested on volumetric attacks cannot be expected to detect protocol-based attacks that exhibit lower traffic rates.

3.4 Sensitivity of Performance to Dataset Choice

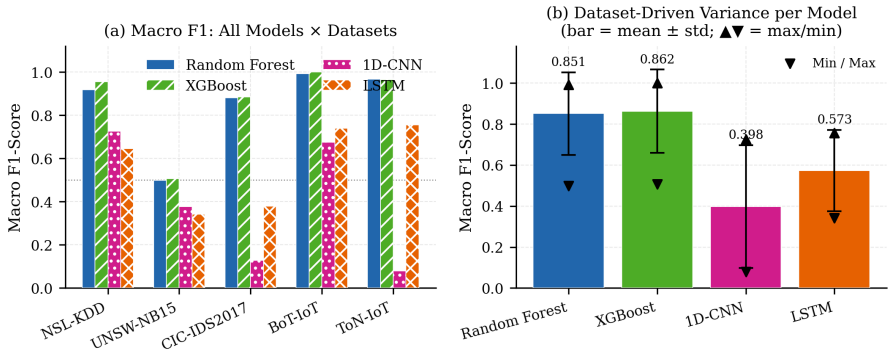


Fig. 4 Macro F1-score for four models across five benchmark datasets. Dataset-driven variance within a fixed model frequently exceeds model-driven variance within a fixed dataset.

Table 2 and Figure 4 present macro-F1 scores for all twenty model–dataset combinations. Several patterns warrant attention.

Dataset choice is the primary source of evaluation variance. The macro-F1 of any fixed model varies substantially across datasets: RandomForest spans 0.4934 (0.9925 on BoT-IoT versus 0.4991 on UNSW-NB15), XGBoost spans 0.4935, CNN spans 0.6457, and LSTM spans 0.4135. This across-dataset spread is an order of magnitude larger than the within-dataset variation across random splits reported in Table 3 (all per-cell standard deviations below 0.025), confirming that dataset selection is a substantially more significant driver of evaluation outcomes than partition choice. The practical implication is that a model claiming state-of-the-art performance evaluated on a single corpus may rank substantially differently when evaluated on a corpus with different structural properties.

Classical ML models outperform deep learning on tabular benchmark features. XGBoost achieves the highest mean macro-F1 across all five datasets

Table 2 Benchmarking Results: Macro F1-Score, Accuracy, and Training Time Across IDS Benchmark Datasets

Dataset	Model	Acc.	Macro-F1	W-F1	Prec.	Rec.	Time (s)
NSL-KDD	Random Forest	0.9946	0.9186	0.9945	0.9607	0.8904	1
	XGBoost	0.9944	0.9547	0.9945	0.9311	0.9825	11
	CNN	0.9460	0.7259	0.9549	0.6967	0.9187	193
	LSTM	0.8740	0.6450	0.8971	0.6214	0.8558	872
UNSW-NB15	Random Forest	0.7491	0.4991	0.7821	0.5256	0.5366	2
	XGBoost	0.6832	0.5065	0.7373	0.5110	0.6321	21
	CNN	0.5992	0.3780	0.6734	0.4199	0.5552	471
	LSTM	0.5641	0.3426	0.6463	0.4020	0.5236	2332
CIC-IDS2017	Random Forest	0.9987	0.8803	0.9986	0.9066	0.8654	112
	XGBoost	0.9988	**0.8859**	0.9988	0.8751	0.9008	253
	CNN	0.7579	0.1288	0.7628	0.1184	0.1874	1167
	LSTM	0.8482	0.3791	0.8859	0.3173	0.8529	56405
BoT-IoT	Random Forest	1.0000	0.9925	1.0000	0.9979	0.9875	133
	XGBoost	1.0000	**1.0000**	1.0000	1.0000	1.0000	82
	CNN	0.9883	0.6755	0.9886	0.6358	0.9380	3462
	LSTM	0.9939	0.7406	0.9941	0.7005	0.9684	14172
ToN-IoT	Random Forest	0.9887	0.9666	0.9888	0.9634	0.9703	1
	XGBoost	0.9889	0.9649	0.9892	0.9538	0.9818	19
	CNN	0.2352	0.0802	0.1519	0.1175	0.1595	28
	LSTM	0.7991	**0.7561**	0.8159	0.7624	0.8197	1062

(0.8624), followed by RandomForest (0.8514). The CNN (0.3977) and LSTM (0.5727) trail considerably. This outcome reflects the nature of benchmark datasets rather than a general inferiority of deep learning for intrusion detection. Benchmark corpora present tabular, flow-level features in which the decision boundaries between attack types are often highly regular and near-axis-aligned, a regime where gradient-boosted trees have a well-documented structural advantage. Deep learning architectures, which are better suited to raw or semi-structured inputs where spatial or temporal structure is meaningful, do not benefit from being applied to pre-aggregated flow statistics. In real-world IoT settings where packet-level or multi-modal inputs are available, this relative performance ranking may reverse. The finding therefore motivates, rather than contradicts, the design guidelines for next-generation datasets presented in Section 4.

BoT-IoT shows extreme structural separability for tree-based models. XGBoost achieves a macro-F1 of 1.0000 on BoT-IoT (confirmed as genuine

near-perfection rather than data leakage; the confusion matrix records only 2 misclassifications across 733,705 test flows, and the identifier-exclusion protocol described in Section 6 was applied throughout). RandomForest achieves 0.9925. By contrast, the CNN achieves 0.6755 and the LSTM 0.7406 on the same dataset, a gap of 0.2549 and 0.2519 respectively between classical and deep approaches. This pronounced ML/DL split is unique among the five corpora and suggests that the feature space of BoT-IoT contains near-perfect decision boundaries exploitable by tree splits but not easily captured by gradient-based sequence learning on tabular inputs. While this may reflect genuine simplicity in the traffic patterns captured, it equally reflects a dataset limitation: real IoT botnet behaviour is considerably more diverse and evasive than what the laboratory captures used in BoT-IoT can represent [15].

UNSW-NB15 is the most discriminating benchmark. All four models achieve their lowest or near-lowest macro-F1 on UNSW-NB15 (RF: 0.4991, XGB: 0.5065, CNN: 0.3780, LSTM: 0.3426), with an average across models of 0.4316. This aligns with the structural properties of the corpus: a wider variety of attack subtypes (10 categories including fine-grained classes such as Worms, Shellcode, and Analysis), more realistic feature overlap between attack families, and a test-set composition deliberately designed to differ from training data. These properties make UNSW-NB15 the most reliable single-corpus stress test among the five benchmarks, despite its limitations regarding IoT-specific scenarios.

Deep learning convergence is unreliable on severely imbalanced corpora. The CNN on CIC-IDS2017 terminated training at 6 epochs (macro-F1: 0.1288) and on ToN-IoT similarly at 6 epochs (macro-F1: 0.0802), with validation loss increasing monotonically from the first epoch in both cases. The LSTM trained for the full 50 epochs on CIC-IDS2017 (56,405 seconds) and achieved a macro-F1 of only 0.3791, marginally better than the CNN but at orders-of-magnitude greater computational cost. These outcomes are not architectural failures per se but rather consequences of

training dynamics in the presence of extreme class imbalance: gradient signal from minority classes is drowned out even under class-weighted loss, and early stopping fires before meaningful learning can occur. They reinforce the importance of the class balance and annotation pillar articulated in the design guidelines.

3.5 Robustness to Data-Split Selection

The results in Table 2 are computed on a single stratified split. To assess whether these values are stable rather than artefacts of a particular partition, the two strongest models — Random Forest and XGBoost — were re-evaluated across five independent stratified 80/20 splits using distinct random seeds. Table 3 reports the resulting mean and standard deviation of macro-F1. The deep models were excluded from this analysis owing to their substantially greater computational cost (the LSTM alone requiring approximately 15.7 hours per run on CIC-IDS2017); as the weakest performers by a wide margin, their exclusion does not affect the paper's comparative conclusions.

Table 3 Macro-F1 Across Five Independent Splits (Mean ± Standard Deviation)

Dataset	Model	Macro-F1 (mean ± std)
NSL-KDD	Random Forest	0.900 ± 0.015
	XGBoost	0.935 ± 0.008
UNSW-NB15	Random Forest	0.576 ± 0.004
	XGBoost	0.585 ± 0.006
CIC-IDS2017	Random Forest	0.864 ± 0.024
	XGBoost	0.902 ± 0.012
BoT-IoT	Random Forest	0.994 ± 0.006
	XGBoost	0.999 ± 0.001
ToN-IoT	Random Forest	0.968 ± 0.001
	XGBoost	0.965 ± 0.002

Two observations follow. First, the standard deviations are small (all below 0.025), confirming that model rankings and the broad performance pattern are stable across

splits rather than fragile to partition choice. Second, and more instructively, the single-split UNSW-NB15 values in Table 2 (RF 0.499, XGBoost 0.507) fall well below the corresponding five-split means (0.576 and 0.585), indicating that the original partition happened to be a comparatively difficult one for this corpus. This gap is itself an illustration of the paper's central methodological argument: single-split evaluations can misrepresent a dataset's difficulty, and repeated-split reporting is necessary for reliable characterisation. Notably, the near-perfect separability of BoT-IoT persists across all five splits (XGBoost 0.999 ± 0.001), reinforcing that its high scores reflect genuine structural separability rather than a fortunate partition.

3.6 Implications for IoT Deployments

Four practical directives emerge from the analysis above:

- **Macro-F1 over accuracy.** Given that every dataset in this study produced accuracy—macro-F1 gaps exceeding 0.05, aggregate accuracy should not serve as a primary evaluation metric for IDS research.
- **Multi-dataset evaluation is essential.** The 0.49-point macro-F1 gap between best and worst datasets for both RF and XGBoost demonstrates that single-corpus evaluations cannot reliably characterise model capability.
- **IoT-specific tactic coverage is currently absent.** None of the five datasets captures the lightweight protocol abuse, multi-stage lateral movement, or encrypted command-and-control behaviour that characterise sophisticated IoT intrusions.
- **Architectural comparisons require dataset-aware framing.** The consistent advantage of classical ML over deep learning observed here is a property of tabular benchmark features, not a general result, and dataset design described in Section 5 is a prerequisite to meaningful architectural comparison in realistic settings.

These observations directly motivate the dataset design principles and testbed roadmap presented in Sections 4 and 5.

4 Design Guidelines for IoT IDS Datasets

The guidelines for constructing intrusion detection datasets relevant to IoT are articulated along six pillars synthesised in Figure 5. Each pillar is motivated by the deficits identified in Section 3 (Figures 1–4) and expressed as a set of requirements whose fulfilment can be verified during curation and before release. The purpose is to improve external validity in IoT settings while preserving ethical and privacy constraints.

The quantitative thresholds attached to each pillar (for example, a minimum encrypted-traffic fraction or a maximum inter-arrival-time divergence) should be understood as *proposed starting-point conventions* rather than as empirically optimal values. They are set at deliberately conservative levels informed by the deficits observed in Section 3 and by values reported in the measurement literature where available and are intended as concrete, checkable defaults that the community can refine as evidence accumulates. Their primary contribution is to convert qualitative desiderata, which prior frameworks already articulate, into verifiable acceptance criteria; the specific numeric levels are secondary and revisable.

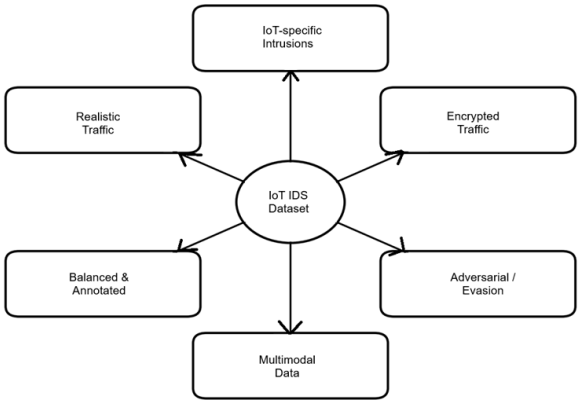


Fig. 5 Design guidelines.

1. **IoT-specific intrusions:** A dataset is expected to capture threats that occur in IoT environments rather than relying primarily on enterprise-centric or legacy attacks. Scenarios should reflect device hijacking, IoT botnet propagation, protocol abuses (e.g., MQTT/CoAP/AMQP/OPC-UA), supply chain and insider activity, and multistage progressions from initial access through lateral movement to command-and-control and impact. Device roles (sensor, actuator, controller, gateway) and lifecycle phases (onboarding, updates, maintenance, failure) are to be encoded so that benign behaviour is represented with comparable fidelity to attacks, with coverage established through a scenario catalogue and run manifests. As a minimum threshold, a dataset should represent at least eight distinct MITRE ATT&CK tactics (or their ICS-ATT&CK equivalents), include at least three IoT-specific protocols, and provide at least one complete multi-stage attack chain from initial access through to impact.
2. **Encrypted traffic:** Given the significant role played by encrypted communications in today's traffic, there is a need for realistic models. While payload encryption needs to be maintained, the following surface properties must be revealed for inspection purposes, with clear annotations denoting the use of encrypted command and control or beaconing in case of orchestration: handshake metadata, JA3/JA3S fingerprints, SNI flags, certificate properties, and time/size sequence. A rule of thumb is that at least 30% of all the traffic must have TLS/DTLS encryption; this percentage takes into account the trend in the adoption of TLS encryption in IoT installations as captured in some recent measurements and forms a realistic lower bound. Furthermore, the KL divergence between the distribution of ciphers used in the dataset and that in some real IoT setup cannot be greater than 0.10.
3. **Adversarial and evasion behaviour:** In order to facilitate a robustness evaluation, the evasion techniques should be incorporated in a systematic manner,

such as packet fragmentation and reordering, jittered time intervals, protocol tunnelling, domain fluxion, polymorphic payloads, and randomized beaconing, with an explicit label based on its evasion purpose and detectable signature surface. This is achieved through the degradation of baselines compared to that of non-evaded variants, along with parameter details. As a minimum requirement, there must be four types of evasion technique variations, with each having no less than 500 labelled instances.

4. **Multimodal observability:** Flow/packet traces should be augmented with protocol connection data (i.e., Zeek logs) and, selectively, with telemetry data at the host level and application levels (i.e., Sysmon/auditd events, controller logs, state changes). The use of NTP time synchronization and stable identifiers for entities is necessary in order to facilitate joining of data across modalities. This should be accompanied by a versioned schema (CSV/Parquet + YAML/JSON metadata) and a few reference scripts for ingestion. Specifically, skew of clocks on collection systems should not exceed 50 ms and joinability of flow records to at least one host level or protocol-level record should be at least 95%.
5. **Class balance and annotation:** Given the heavy-tailed distributions observed in Section 3, both a natural release (retaining organic skew) and a balanced derivative (with documented sampling or weighting) are recommended. Labels should be multidimensional, capturing family, stage, technique/TTP, target role, and success status, aligned to ATT&CK (and ICS-ATT&CK where applicable), with annotation provenance recorded and stratified interannotator checks reported. Two quantitative targets follow directly from the imbalance analysis: no single class should account for more than 40% of total flows in the balanced derivative release, and every evaluable class should contain a minimum of 500 samples in the test

partition. Classes below this floor, such as Heartbleed (2 test samples) and Infiltration (7 test samples) observed in CIC-IDS2017, should be explicitly excluded from macro-averaged reporting or designated as held-out edge cases.

6. **Realistic traffic dynamics:** Diurnal patterns, burstiness, concurrency, session re-use and edge-to-cloud flows must all be included in the background workload, along with scripted behavior mixed in with natural IoT device activity, and, insofar as is possible, hardware-in-the-loop devices should be used to account for timing behavior not realistically modeled through virtualisation. These three distributional requirements specify this criterion. Firstly, the KL divergence between the inter-arrival time distribution in the dataset and a real traffic distribution should not exceed 0.15. Secondly, there must be a statistically significant difference (at $p < 0.05$ under the Kolmogorov-Smirnov test) between flow volumes during peak hours and non-peak hours. Thirdly, the coefficient of variation of concurrently active sessions must be larger than 0.3.

4.1 Integration and Verification

The six pillars in Figure 5 are interdependent: encrypted traffic and evasion must go hand-in-hand in multi-stage setups; multimodal observability requires coordinated time and consistent identifiers; and class balancing cannot compromise the temporal dynamics that are crucial for realism. The operational pipeline, from topology to workloads to playbooks to collection and labelling, is covered in Section 5 and Figure 6, as well as the manifests and schemas necessary to meet the needs of reproducibility downstream.

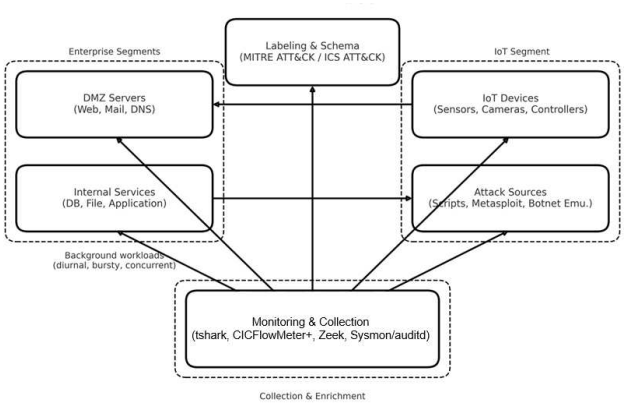


Fig. 6 Testbed roadmap / topology.

5 Testbed Roadmap for Dataset Construction

Dataset creation is structured as an end-to-end pipeline, illustrated in Figure 6. This involves exercising the segmented network topology using background tasks and controlled attacks; collecting traffic and telemetry using a single collection infrastructure; and generating multi-dimensional labels using a defined labelling schema.

1. **Topology and segmentation:** A three-segment layout (web/mail/DNS in the DMZ, database/file/application for business backends, sensors/cameras/controllers representing the IoT devices) is used to reflect an enterprise-IoT architecture in common usage. DMZ services act as entry points to the environment, backend services as business operations, and the devices present lightweight protocols and edge-to-cloud channels. VLANs or virtual switches isolate segments from each other, while designated segments are made to allow for some form of reconnaissance, initial entry, lateral movement, and C2 communication. This provides realistic attack vectors, alongside benign behavior, on which to test detection.
2. **Background workloads and traffic dynamics:** A combination of scripted user activity (web surfing, email, CRUD on databases, synchronisation of files) and device-specific background noise (telemetry bursts, keep-alive packets, update

requests) is used to generate benign behaviour. Scheduled job mixes and randomised inter-arrival times are used to create diurnal patterns, traffic bursts, concurrency, and connection reuse. Where possible, hardware-in-the-loop is used to reflect the impact of clock and jitter behaviour on network traffic, which is lost when using virtual devices.

3. **Attack sources and playbooks:** The Attack Sources block runs parameterized playbooks for IoT-relevant threats: propagation of IoT botnets, exploit/abuse of MQTT/CoAP, brute-force/credential-stuffing, DoS/DDoS, exfiltration, ransomware staging, and multistage or APT-style chains (initial access → privilege escalation → lateral movement → C2 → impact). Detector robustness can be tested by enabling various forms of evasive manoeuvres (fragmentation/reordering, timing jitter, tunnelling, domain flux, polymorphic payload, randomised beaconing). Each run is documented in a run manifest for reproducibility, including seeds, parameters, timestamps, affected roles, and expected impacts.
4. **Monitoring and collection layer:** All sections communicate their traffic and telemetry with the common Monitoring & Collection layer. Packet capture (tshark etc.), flow feature generation (CICFlowMeter+ etc.), protocol/connection metadata (Zeek etc.), and host-level telemetry (Sysmon/auditd, etc. on select nodes) take place. Global time synchronisation (NTP) is guaranteed, and a set of steady entities (host, interface, process) and session/run identifiers allows reliable joins across different modes. Append-only data storage is employed, with immutable manifests that hold information on software versions and checksums in use.
5. **Labelling and ontology alignment:** The Labelling & Schema block generates ground truth. Labels are multi-dimensional (attack family, stage, technique/TTP, target role, success flag, scenario ID) and mapped to MITRE ATT&CK (and, where applicable, ICS-ATT&CK). Ground truth assertions are performed at the boundary of scenarios by a lightweight event logger; post-hoc validators reconcile

injected events against traces (Zeek notices, authentication logs, etc.) to report failures to inject or inject fragments. Label provenance is recorded (auto-asserted, scripted marker, or human-reviewed), and stratified inter-annotator agreement is reported for human-reviewed parts.

6. **Data products and splits:** Three categories of artefacts are prepared for release:

   - *Raw:* pcaps, Zeek logs, host logs, and configuration manifests;
   - *Derived:* flow tables (CSV/Parquet) and joined multimodal tables with explicit join keys (time, entity, session);
   - *Schemas & Code:* versioned YAML/JSON schemas and reference notebooks that demonstrate ingestion, joining, and baseline training.

   Two split philosophies are provided: a natural split that preserves organic skew and a balanced derivative with documented sampling or weighting for classifier stress tests. Scenario-focused subsets (e.g., encrypted-C2, lateral-movement) are additionally curated to enable targeted evaluation.

7. **Quality assurance and validation:** The quality controls operate on three levels. Realism is verified via distributional diagnostics (flow lengths, interarrival times, concurrent session counts) and tests for diurnality, burstiness, session reuse; traffic too uniform is reported. Coverage is checked against the expected catalogue of scenarios (class thresholds per class, counts of tactics and techniques). Integrity is enforced through bounds on clock skew and packet loss, consistency of labels and telemetry, and tests for data leakage between splits. Every release is delivered with a machine-readable validation report and scripts to regenerate it.

8. **Privacy and release hygiene:** The header and identifier are anonymised, and the domain may optionally be redacted, and sensitive payloads may be truncated. Host logs are processed to remove personal information while retaining the necessary fields for security analysis (timestamps, event types, file hashes, etc.). Licenses,

acceptable use and an ethics statement are provided along with a reporting channel to submit findings.

9. **Orchestration and reproducibility:** The orchestration scripts are presented end-to-end to automatically orchestrate topology re-design, workload scheduling, scenario execution, capture, post-processing and packaging. The result should be reproducible and auditable using a run manifest (seeded randomness, software versions, parameters) and hash-stamped artefacts. Automatic, constant checks are in place to ensure that new runs do not exceed the given tolerance bands for realism and coverage.

Together, these nine stages operationalise the design principles of Section 4 and address the structural, statistical, and semantic gaps identified in Section 3.

6 Pilot Dataset: Demonstrating Guideline Compliance

The preceding sections identified empirical gaps in existing IDS corpora and proposed six quantitative design guidelines to address them. This section demonstrates that these guidelines are largely achievable by presenting a pilot dataset constructed from the virtualised enterprise-IoT testbed described in Section 5. The pilot is a proof of concept: it is not proposed as a replacement for existing benchmarks, nor as a production-scale training corpus. Its purpose is to provide concrete evidence that the design principles of Section 4 can be realised within a modest research infrastructure, and that the resulting data exhibits structural properties absent from current corpora.

6.1 Testbed Configuration and Capture Methodology

The testbed comprises ten virtual machines distributed across four isolated internal networks: *enterprise* (192.168.10.0/24), *IoT* (192.168.20.0/24), *attacker* (192.168.30.0/24), and *management* (192.168.99.0/24). The enterprise segment hosts

an Apache/DVWA web server, a Flask REST application server with an intentionally vulnerable endpoint, a MySQL database server, and a workstation running a randomised background-traffic generator. The IoT segment comprises three sensor nodes publishing MQTT and MQTT-over-TLS telemetry and issuing CoAP requests to a gateway running a Mosquitto broker (plain on port 1883 and TLS on port 8883) and a CoAP server. A Kali Linux node executes controlled attack playbooks. A dedicated management-and-capture node spans all four networks with promiscuous-mode interfaces, running Zeek (four workers, one per segment), tshark for packet capture, and Chrony as a stratum-1 NTP server synchronising all nodes to within 10 ms.

Per-flow features were extracted using NFStream [40] with statistical analysis enabled, producing 86 features per flow spanning bidirectional and directional packet and byte counts, flow duration, packet-size statistics, inter-arrival-time statistics, and TCP flag counts. Approximately 24 hours of benign background traffic were captured, followed by a series of targeted attack sessions and a single multi-stage attack chain. All attack traffic was generated within a fully isolated virtual testbed with no connection to external networks or production systems, and no human-subject or personal data was collected at any stage; the work therefore posed no risk to external parties or systems. Each session was captured and labelled independently, with ground-truth labels assigned by session boundary rather than inferred post hoc from addresses or payload signatures.

6.2 Deduplication and Data Integrity

Initial processing revealed a substantial proportion of exact-duplicate feature vectors, concentrated in the volume-based attack classes. Volume-based attacks exhibited the highest duplication: 97.2% of PortScan flows were exact duplicates, reflecting the inherently repetitive nature of scan traffic, in which large numbers of probes share identical flow-level characteristics. Benign IoT telemetry was similarly repetitive (88.8%),

driven by periodic MQTT keepalives and CoAP polling with near-identical signatures. A per-class deduplication step was therefore incorporated into the pipeline, removing exact-duplicate feature vectors within each class while preserving at least one representative of every distinct signature. This finding has methodological significance beyond the present dataset: raw flow counts can substantially overstate the effective sample diversity of an IDS dataset, a consideration rarely reported for existing benchmark corpora and one that connects directly to the accuracy–macro-F1 critique of Section 3.

The multi-stage attack chain (described below) required special handling. Its final stage, a denial-of-service flood, generated approximately 65,000 near-duplicate flows that would have swamped the sequential signal of the chain's earlier stages. The chain flows were therefore truncated at the onset of the denial-of-service stage, retaining only the sequential core (reconnaissance through exfiltration); this preserves the multi-stage progression of interest without diluting it with high-volume flood traffic already represented by the dedicated DoS class.

6.3 Pilot Dataset Profile

Table 4 Deduplicated Pilot Dataset (v2): Class Distribution and ATT&CK Alignment

Class	ATT&CK Tactic(s)	Flows
BruteForce	Credential Access	10,829
BENIGN	—	9,895
DoS	Impact	5,865
SQLInjection	Execution	1,672
MQTTAttack	Command & Control	957
PortScan	Reconnaissance	522
MultiStageAttack	Recon. → Discovery → Cred. Access → Lateral Mvmt. → Exfil.	140
Total		29,880

Table 4 summarises the final deduplicated dataset of 29,880 unique flows across seven classes. Six single-tactic attack classes span the ATT&CK tactics of Reconnaissance, Credential Access, Execution, Impact, and Command and Control alongside

a benign class. The seventh class, MultiStageAttack, is a single chained intrusion sequence described in Section 6.4. Three IoT-relevant application protocols are represented: MQTT, CoAP, and MQTT-over-TLS. The MQTTAttack class, capturing MQTT-specific command-and-control behaviour, and the encrypted MQTT-over-TLS telemetry have no counterpart in any of the five benchmark corpora evaluated in Section 3.

6.4 Multi-Stage Attack Chain

A distinguishing feature of the pilot is a genuine multi-stage attack chain, executed as a single continuous sequence rather than as independent attack types. Beginning from the attacker node, the chain proceeds through five ATT&CK tactics: (i) *Reconnaissance* — scanning the enterprise subnet; (ii) *Discovery* — service and version enumeration of identified hosts; (iii) *Credential Access* — an SSH brute-force attempt against a foothold host; (iv) *Lateral Movement* — pivoting from the compromised foothold to reach the database server on an otherwise unreachable path; and (v) *Collection and Exfiltration* — retrieving a sensitive file from the database server and transferring it toward the attacker segment. Stage transitions were timestamped at capture time, providing unambiguous ground truth for the tactic sequence. This chained progression, absent from all five benchmark corpora and representing attacks only as isolated single-stage events, directly addresses the multi-stage coverage gap identified in Section 3 and realises the complete initial-access-through-impact sequence required by Guideline 1.

6.5 Leakage Audit

To ensure that model performance reflects genuine class structure rather than artefactual shortcuts, a two-part leakage audit was conducted. First, all flow-identifier fields (source and destination IP addresses, MAC addresses, ports, and NFStream flow

identifiers) were removed prior to training, since these can encode the experimental topology. Second, the discriminative power of each individual feature was assessed by training a depth-limited decision tree on that feature alone. The most predictive single feature achieved 0.84 accuracy, and no feature exceeded the 0.90 threshold that would indicate single-feature leakage. After deduplication, the residual exact-duplicate rate was 2.5%, confirming that train–test contamination through duplicated flows was effectively eliminated. The same identifier-exclusion protocol applied to BoT-IoT in Section 3 left its near-perfect separability intact, indicating that those high scores arise from genuine feature separability rather than identifier leakage.

6.6 Model Evaluation

To assess whether the pilot supports meaningful model evaluation, Random Forest and XGBoost were trained and evaluated using the identical methodology to Section 3 (80/20 stratified split, standardisation fitted on the training partition only, balanced class weighting). Following Guideline 5, which prescribes that classes below the 500-flow support floor be excluded from macro-averaged reporting, the MultiStageAttack class (140 flows) is excluded from the primary macro-average and reported separately; it is retained in the dataset to demonstrate chained-attack capability rather than to provide classification volume. Results over the six evaluable classes are reported in Table 5.

Table 5 Model Performance on the Pilot Dataset (Six Evaluable Classes)

Model	Accuracy	Macro-F1	Macro-F1 (5-seed)
Random Forest	0.9349	0.8213	0.822 ± 0.004
XGBoost	0.9277	0.8082	0.808 ± 0.006

The six-class macro-F1 of approximately 0.82 situates the pilot between the most difficult benchmark corpus (UNSW-NB15, macro-F1 0.50) and the moderately separable corpora (CIC-IDS2017, macro-F1 0.88), and well below the near-perfectly separable BoT-IoT (macro-F1 1.00); the pilot therefore presents a realistic classification challenge. Per-class results (Random Forest) were: BruteForce 0.979, DoS 0.958, SQLInjection 0.919, BENIGN 0.927, MQTTAttack 0.758, and PortScan 0.387. PortScan, with the fewest unique flows (522) and high intra-class variability arising from randomised scan parameters, achieved the lowest per-class score, mirroring precisely the minority-class detection difficulty documented for benchmark corpora in Section 3. Evaluated separately, the MultiStageAttack class achieved an F1 of 0.21, which was expected given its minimal support, confirming that its value lies in demonstrating tactic sequencing rather than in classification benchmarking.

6.7 Guideline Compliance Assessment

Table 6 scores the pilot dataset against each of the six proposed guidelines, alongside the five benchmark corpora evaluated in Section 3. It is noted that this comparison illustrates the motivating gap rather than independently validating the guidelines against an external ground truth: the guidelines were themselves derived from the deficits observed in these same corpora, and the comparison should be read as a structured gap analysis rather than as an independent ranking of dataset quality. Each mark for the pilot is substantiated by a quantity reported in the preceding subsections.

The pilot fully satisfies four of the six guidelines and partially satisfies two, with each mark traceable to a reported quantity:

- **G1 (✓)**: three IoT protocols (MQTT, CoAP, MQTT-over-TLS); eight ATT&CK tactics across the seven classes; and one complete multi-stage chain (Section 6.4).
- **G2 (P)**: 623 TLS-encrypted flows are present (2.1% of the dataset), demonstrating that the pipeline captures genuine encrypted traffic; the proportion remains below

Table 6 Guideline Compliance: Pilot Dataset vs Existing Benchmarks (✓= threshold met, P = partially met, × = not met)

Guideline	NSL-KDD	UNSW-NB15	CIC-IDS2017	BoT-IoT	ToN-IoT	Pilot
G1: IoT protocols (≥3), tactics (≥8), chain (≥1)	×	×	×	P	P	✓
G2: Encrypted traffic (≥30% TLS)	×	×	×	×	×	P
G3: Adversarial/evasion variants (≥4)	×	×	P	×	×	P
G4: Multimodal observability (skew ≤50ms; join ≥95%)	×	×	×	×	×	✓
G5: Class support (≥500) and max share (≤40%)	P	P	×	P	P	✓
G6: Realistic traffic dynamics (randomised intervals)	×	×	P	×	×	✓
Fully met	0	0	0	0	0	**4**
Partially met	1	2	2	2	3	**2**

the 30% threshold, which a fuller encrypted-traffic capture in the full dataset will address.

- **G3 (P)**: multiple DoS variants (SYN, ACK, FIN, RST, and UDP floods; slowloris) and SQL-injection variants are present, but timing-based evasion and protocol tunnelling are not yet included.
- **G4 (✓)**: inter-node clock skew held below 10 ms via Chrony; flow records are joinable to Zeek connection logs across all segments.
- **G5 (✓)**: all six evaluable classes exceed the 500-flow support floor, and the largest class (BruteForce) constitutes 36.2% of the dataset, below the 40% ceiling.
- **G6 (✓)**: benign traffic is generated with randomised inter-arrival intervals across enterprise and IoT services.

No existing benchmark fully satisfies more than a single guideline within this comparison; it is reiterated that this partly follows from the guidelines having been derived from the deficits of these same corpora.

6.8 Distinctive Characteristics and Limitations

Four properties of the pilot are absent from all five benchmarks: authentic IoT protocol behaviour (MQTT, CoAP, and MQTT-over-TLS generated by real broker and server implementations); encrypted traffic with recoverable flow-level metadata; a genuine multi-stage attack chain with capture-time tactic labelling; and multi-segment attack paths traversing a forwarding node. Together these give the pilot a structural profile that none of the surveyed corpora provides.

Three limitations define the transition from pilot to the full dataset. First, although encrypted traffic is now present, its proportion (2.1%) remains well below the 30% target; substantially expanding TLS-based telemetry is the highest-priority extension. Second, the deduplicated scale (29,880 unique flows) is modest and suited to demonstrating guideline compliance rather than large-scale model training. Third, evasion coverage (G3) is partial, lacking timing-based and tunnelling variants. These extensions are planned within the full dataset-construction effort and will be evaluated against the same compliance criteria presented here. Notwithstanding these limitations, the pilot provides the first empirical demonstration that the proposed guidelines are realisable within a modest testbed, and that the resulting dataset supports stable, reproducible, and realistically challenging model evaluation.

7 Conclusion and Future Work

Intrusion detection systems remain critical to IoT security, but their reliability is undermined by the inadequacies of current benchmark datasets. This study examined widely used intrusion detection corpora through an IoT-aware lens and highlighted

structural, statistical and semantic gaps that limit external validity. It was shown, with reference to the accompanying figures, that heterogeneous feature schemas, severe class imbalance, incomplete coverage of adversary tactics (notably persistence, lateral movement, and encrypted C2), and limited multimodality collectively bias model evaluation; moreover, performance was observed to be highly sensitive to the dataset itself rather than the learning architecture alone. In response, a set of design guidelines was articulated and a testbed-driven roadmap was outlined to enable the construction of datasets that better reflect operational IoT environments.

The guidelines emphasise IoT-specific intrusions, encrypted communications, adversarial/evasion behaviour, multimodal observability, balanced and well-provenanced labels and realistic traffic dynamics. The proposed pipeline organises these requirements into a reproducible process, from topology and background workloads to attack playbooks, collection, labelling, validation and privacy-preserving release. Together, these measures are intended to improve the credibility, comparability, and practical utility of future IDS evaluations in IoT settings.

Several limitations should be acknowledged. The benchmarking study, while spanning five corpora and four model families, represents a single experimental configuration; comprehensive, community-wide benchmarks adhering to the proposed principles remain to be produced. The pilot dataset of Section 6 demonstrates the feasibility of a subset of the guidelines but is modest in scale and does not yet satisfy the encrypted-traffic criterion. Future work is therefore motivated along three axes: (i) extension of the pilot into a full, publicly releasable IoT dataset, including TLS-encrypted traffic and a broader evasion playbook, with subsequent extension toward industrial and critical-infrastructure protocols; (ii) development of shared schemas, validation reports, and split protocols to standardise evaluation; and (iii) broader engagement on privacy and ethics to support open, yet responsible dissemination. Through these steps, more reliable and transferable IDS research can be enabled for IoT.

List of Abbreviations

AMQP: Advanced Message Queuing Protocol; APT: Advanced Persistent Threat; ATT&CK: Adversarial Tactics, Techniques and Common Knowledge; C2: Command and Control; CNN: Convolutional Neural Network; CoAP: Constrained Application Protocol; DDoS: Distributed Denial of Service; DL: Deep Learning; DoS: Denial of Service; F1: F1-score; ICS: Industrial Control System; IDS: Intrusion Detection System; IoT: Internet of Things; KL: Kullback–Leibler; LSTM: Long Short-Term Memory; ML: Machine Learning; MQTT: Message Queuing Telemetry Transport; NIDS: Network Intrusion Detection System; NTP: Network Time Protocol; PIAT: Packet Inter-Arrival Time; RF: Random Forest; TLS: Transport Layer Security; TTP: Tactics, Techniques and Procedures; U2R: User-to-Root; XGB: XGBoost.

Declarations

Ethics approval and consent to participate

Not applicable. This study did not involve human participants, human data, human tissue, or animals. All experiments were conducted within a fully isolated virtual testbed with no connection to external networks or production systems.

Consent for publication

Not applicable. This manuscript does not contain any individual person's data in any form.

Availability of data and materials

The experimental pipeline supporting the conclusions of this article, including all preprocessing scripts, model-training notebooks and configuration files, is available in the GitHub repository: https://github.com/ifrahnaqash/ids-benchmark-experiments.

The five benchmark datasets analysed during the current study are publicly available from their original sources and are cited in the reference list [35–39]. The pilot dataset generated during the current study is not yet publicly available, as it forms part of an ongoing full dataset construction effort and requires anonymisation of host-level identifiers prior to release, but is available from the corresponding author on reasonable request.

Competing interests

The authors declare that they have no competing interests.

Funding

The authors declare that no funds, grants, or other support were received during the preparation of this manuscript.

Authors' contributions

IS conceived the study, designed and implemented the benchmarking experiments, testbed, and pilot dataset, and drafted the manuscript. RNM supervised the research, contributed to the study design and validation, and reviewed and edited the manuscript. All authors read and approved the final manuscript.

Acknowledgements

The authors gratefully acknowledge the Department of Computer Science and Engineering, National Institute of Technology Srinagar, for providing the academic environment and computational facilities that enabled this work.

References

[1] Rao NKS, et al. Transformative applications of IoT in diverse industries: a mini review. Malays J Sci Adv Technol. 2024;:130–140. doi:10.56532/mjsat.v4i2.292.

[2] Li C, Wang J, Wang S, Zhang Y. A review of IoT applications in healthcare. Neurocomputing. 2023;565:127017. doi:10.1016/j.neucom.2023.127017.

[3] Taha AEM, Rashwan AM, Hassanein HS. Secure communications for resource-constrained IoT devices. Sensors. 2020;20(13):3637. doi:10.3390/s20133637.

[4] Bradbury M, Jhumka A, Watson T. Information management for trust computation on resource-constrained IoT devices. Future Gener Comput Syst. 2022;135:348–363. doi:10.1016/j.future.2022.05.004.

[5] Canavese D, Mannella L, Regano L, Basile C. Security at the edge for resource-limited IoT devices. Sensors. 2024;24(2):590. doi:10.3390/s24020590.

[6] Larriva-Novo X, Villagrá VA, Vega-Barbas M, Rivera D, Rodrigo MS. An IoT-focused intrusion detection system approach based on preprocessing characterization for cybersecurity datasets. Sensors. 2021;21(2):656. doi:10.3390/s21020656.

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

[8] Alshammari A, Aldribi A. Apply machine learning techniques to detect malicious network traffic in cloud computing. J Big Data. 2021;8(1). doi:10.1186/s40537-021-00475-1.

[9] Tavallaee M, Bagheri E, Lu W, Ghorbani AA. A detailed analysis of the KDD CUP 99 data set. In: 2009 IEEE Symposium on Computational

765 Intelligence for Security and Defence Applications. Ottawa; 2009. p. 1–6.
766 doi:10.1109/cisda.2009.5356528.

767 [10] Moustafa N, Slay J. UNSW-NB15: a comprehensive data set for network intrusion
768 detection systems (UNSW-NB15 network data set). In: 2015 Military Communi-
769 cations and Information Systems Conference (MilCIS). Canberra; 2015. p. 1–6.
770 doi:10.1109/milcis.2015.7348942.

771 [11] Sharafaldin I, Gharib A, Lashkari AH, Ghorbani AA. Towards a reliable
772 intrusion detection benchmark dataset. Softw Netw. 2017;2017(1):177–200.
773 doi:10.13052/jsn2445-9739.2017.009.

774 [12] Sharafaldin I, Lashkari AH, Ghorbani AA. Toward generating a new intrusion
775 detection dataset and intrusion traffic characterization. In: Proceedings of the 4th
776 International Conference on Information Systems Security and Privacy. Funchal;
777 2018. p. 108–116. doi:10.5220/0006639801080116.

778 [13] Ferriyan A, Thamrin AH, Takeda K, Murai J. Generating network intrusion
779 detection dataset based on real and encrypted synthetic attack traffic. Appl Sci.
780 2021;11(17):7868. doi:10.3390/app11177868.

781 [14] Qaddos A, Yaseen MU, Al-Shamayleh AS, Imran M, Akhunzada A, Alharthi
782 SZ. A novel intrusion detection framework for optimizing IoT security. Sci Rep.
783 2024;14(1). doi:10.1038/s41598-024-72049-z.

784 [15] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. Towards the develop-
785 ment of realistic botnet dataset in the Internet of Things for network foren-
786 sic analytics: Bot-IoT dataset. Future Gener Comput Syst. 2019;100:779–796.
787 doi:10.1016/j.future.2019.05.041.

[16] Alsaedi A, Moustafa N, Tari Z, Mahmood A, Anwar A. TON_IoT telemetry dataset: a new generation dataset of IoT and IIoT for data-driven intrusion detection systems. IEEE Access. 2020;8:165130–165150. doi:10.1109/access.2020.3022862.

[17] Le TTH, Kim H, Kang H, Kim H. Classification and explanation for intrusion detection system based on ensemble trees and SHAP method. Sensors. 2022;22(3):1154. doi:10.3390/s22031154.

[18] Layeghy S, Gallagher M, Portmann M. Benchmarking the benchmark — comparing synthetic and real-world network IDS datasets. SSRN Electron J. 2022. doi:10.2139/ssrn.4141050.

[19] Grover M, Pushparajesh V, Borade MR, Bhuvaneshwari KS, Gupta S, Divya K. Innovative IDS model incorporating machine learning and feature reduction mechanisms. In: 2024 IEEE 4th International Conference on ICT in Business Industry & Government (ICTBIG). Indore; 2024. p. 1–6. doi:10.1109/ictbig64922.2024.10911690.

[20] Kasongo SM, Sun Y. Performance analysis of intrusion detection systems using a feature selection method on the UNSW-NB15 dataset. J Big Data. 2020;7(1). doi:10.1186/s40537-020-00379-6.

[21] Talukder MA, et al. Machine learning-based network intrusion detection for big and imbalanced data using oversampling, stacking feature embedding and feature extraction. arXiv preprint. 2024. doi:10.48550/arxiv.2401.12262.

[22] Kwon D, Neagu RM, Rasakonda P, Ryu JT, Kim J. Evaluating unbalanced network data for attack detection. In: Proceedings of the 2023 on Systems and Network Telemetry and Analytics. New York; 2023. p. 23–26.

doi:10.1145/3589012.3594898.

[23] Gharib A, Sharafaldin I, Lashkari AH, Ghorbani AA. An evaluation framework for intrusion detection dataset. In: 2016 International Conference on Information Science and Security (ICISS). Pattaya; 2016. doi:10.1109/icissec.2016.7885840.

[24] Sanober I, Mir RN. Toward enhanced intrusion detection: a framework for modern and realistic benchmark datasets. In: Lecture Notes in Networks and Systems. Singapore: Springer; 2025. p. 1–11. doi:10.1007/978-981-97-7178-3_1.

[25] Park T, Abuzainab N, Saad W. Learning how to communicate in the Internet of Things: finite resources and heterogeneity. IEEE Access. 2016;4:7063–7073. doi:10.1109/access.2016.2615643.

[26] Albulayhi K, Al-Haija QA, Alsuhibany SA, Jillepalli AA, Ashrafuzzaman M, Sheldon FT. IoT intrusion detection using machine learning with a novel high performing feature selection method. Appl Sci. 2022;12(10):5015. doi:10.3390/app12105015.

[27] Sudyana D, et al. Quality analysis in IDS dataset: impact on model generalization. In: 2024 IEEE Conference on Communications and Network Security (CNS). San Antonio; 2024. p. 1–6. doi:10.1109/cns62487.2024.10735570.

[28] Ferrag MA, Friha O, Hamouda D, Maglaras L, Janicke H. Edge-IIoTset: a new comprehensive realistic cyber security dataset of IoT and IIoT applications for centralized and federated learning. IEEE Access. 2022;10:40281–40306. doi:10.1109/ACCESS.2022.3165809.

[29] Neto ECP, Dadkhah S, Ferreira R, Zohourian A, Lu R, Ghorbani AA. CICIoT2023: a real-time dataset and benchmark for large-scale attacks in IoT environment. Sensors. 2023;23(13):5941. doi:10.3390/s23135941.

[30] Garcia S, Parmisano A, Erquiaga MJ. IoT-23: a labeled dataset with malicious and benign IoT network traffic. Zenodo. 2020. doi:10.5281/zenodo.4743746.

[31] Elam M, Mink D, Bagui SS, Plenkers R, Bagui SC. Introducing UWF-ZeekData24: an enterprise MITRE ATT&CK labeled network attack traffic dataset for machine learning/AI. Data. 2025;10(5):59. doi:10.3390/data10050059.

[32] Booij TM, Chiscop I, Meeuwissen E, Moustafa N, Hartog FTH. TON_IoT: the role of heterogeneity and the need for standardization of features and attack types in IoT network intrusion data sets. IEEE Internet Things J. 2021;9(1):485–496. doi:10.1109/jiot.2021.3085194.

[33] Fatima M, Rehman O, Ali S, Niazi MF. ELIDS: ensemble feature selection for lightweight IDS against DDoS attacks in resource-constrained IoT environment. Future Gener Comput Syst. 2024;159:172–187. doi:10.1016/j.future.2024.05.013.

[34] Alhowaide A, Alsmadi I, Tang J. Towards the design of real-time autonomous IoT NIDS. Cluster Comput. 2021;26(5):2489–2502. doi:10.1007/s10586-021-03231-5.

[35] Canadian Institute for Cybersecurity. NSL-KDD dataset. University of New Brunswick. 2009. https://www.unb.ca/cic/datasets/nsl.html. Accessed 20 Jul 2026.

[36] Moustafa N, Slay J. The UNSW-NB15 dataset. UNSW Canberra. 2015. https://research.unsw.edu.au/projects/unsw-nb15-dataset. Accessed 20 Jul 2026.

[37] Canadian Institute for Cybersecurity. CIC-IDS2017 dataset. University of New Brunswick. 2017. https://www.unb.ca/cic/datasets/ids-2017.html. Accessed 20 Jul 2026.

[38] Koroniotis N, Moustafa N, Sitnikova E, Turnbull B. The Bot-IoT dataset. UNSW Canberra. 2019. https://research.unsw.edu.au/projects/bot-iot-dataset. Accessed 20 Jul 2026.

[39] Moustafa N. The TON_IoT datasets. UNSW Canberra. 2020. https://research.unsw.edu.au/projects/toniot-datasets. Accessed 20 Jul 2026.

[40] Aouini Z, Pekar A. NFStream: a flexible network data analysis framework. Comput Netw. 2022;204:108719. doi:10.1016/j.comnet.2021.108719.