Methodology for building realistic network intrusion detection datasets: a case study with the Energy-Based Flow Classifier.

Eduardo Rodrigues de Oliveira *†,Robson de Oliveira Albuquerque †‡, João José Costa Gondim †
University of Brasilia†
Catholic University of Brasilia‡
Email: eduardo@oliv.com.br*†, robson@redes.unb.br†‡, gondim@unb.br†

***Abstract*—Machine Learning (ML)-based Network Intrusion Detection Systems (NIDS) often show excellent results on synthetic benchmarks, but their effectiveness in real networks is limited by distributional mismatches and weak labeling. Popular datasets such as UNSW-NB15, CIC-IDS2017, and TON_IOT diverge from production traffic and lack reproducible methodologies. Although efforts like HIKARI-2021 emphasized encrypted traces and ID2T improved attack injection, a unified pipeline addressing realism, reproducibility, and temporal validation remains absent. This paper proposes a methodology for constructing realistic, flow-based datasets tailored to anomaly detection. It integrates continuous traffic collection with controlled attacks, assisted and auditable labeling, and temporal splits with feature-stability diagnostics. We validate the pipeline with the Energy-based Flow Classifier (EFC), trained in one-class mode on benign flows and operated by a calibrated threshold. On real traffic, a fixed quantile of 0.95 maintains high recall (about 0.94 to 0.95) with false positive rate of 4.9% on T2 and 8.3% on T3; tuning with F-beta (beta=0.5) or a constrained objective reduces false positives under temporal drift with minor loss in recall. The resulting artifacts enable interpretable, auditable, and drift-aware evaluation in operational settings.**

***Keywords*—NIDS, EFC, Anomaly Detection, Flow-based Datasets, Assisted Labeling, Unsupervised Learning.**

I. Introduction

Machine-learning (ML) Network Intrusion Detection Systems (NIDS) are widely studied, yet their strong laboratory results rarely translate into the same effectiveness in production. A key reason is the distributional mismatch between benign traffic in synthetic benchmarks and real network environments, which undermines model generalization and calls into question the operational validity of results obtained on synthetic datasets [1], [2].

979-8-3315-7075-0/25/$31.00 ©2025 IEEE

Beyond distributional shift, dataset construction itself remains a bottleneck. Publicly available datasets are often outdated, insufficiently documented, weak on labeling methodology, or incomplete with respect to modern conditions such as pervasive encryption. HIKARI-2021 explicitly recognized these issues by specifying content and process requirements for dataset creation, including the publication of build procedures and evidence of incidental anomalies found during background capture—facts that underscore the need for assisted, auditable labeling rather than a purely synthetic approach [3].

Synthetic datasets and attack-injection pipelines are valuable when they respect the statistical properties of the background traffic and actively avoid known defects (e.g., predictable patterns, unrealistic cleanness). Tools such as ID2T moved the field forward by replicating background statistics when generating attacks and detecting defects through dedicated quality checks, improving realism and auditability during dataset creation [4]. At the same time, operating at the flow level (e.g., NetFlow/IPFIX) remains a pragmatic choice that reduces storage and preserves essential communication semantics for NIDS research and deployment.

Given these constraints, anomaly-based, one-class strategies are especially attractive for transfer to new domains. In this work we adopt the *Energy-based Flow Classifier* (EFC) as a case study. EFC models the distribution of benign flows using inverse statistics (inverse Potts model), assigns an *energy* to each flow, and flags anomalies whose energy exceeds a learned threshold. It is interpretable (“white box”) and operates on discretized features, enabling training with benign-only data and facilitating cross-domain use [5].

We propose a reproducible methodology for building realistic, flow-based datasets geared to unsupervised anomaly detection and validated temporally. Specifically, it (i) implements continuous, realistic traffic collection paired with controlled and reproducible attack injection, (ii) adopts assisted and auditable labeling that correlates attack plans with IDS/flow logs and isolates unplanned anomalies for review, (iii) enforces temporally prospective splits (T1/T2/T3) and publishes feature-stability diagnostics to mitigate leakage and quantify drift, (iv) releases the process — scripts, configurations, anonymization settings, and quality reports — alongside the data for end-to-end reproducibility, and (v) is evaluated with EFC trained on benign flows, demonstrating improved robustness under temporal drift and clearer operational interpretability. This pipeline directly addresses the realism–reproducibility trade-off and provides an auditable path from raw capture to deployable anomaly detectors.

II. Related Work and Research Gap

Rigorous evaluation of ML-based NIDS critically depends on dataset quality and representativeness. Quantitative evidence shows that the benign component in synthetic benchmarks (e.g., UNSW-NB15, CIC-IDS2017, TON_IOT) statistically diverges from production traffic, limiting the generalization of models trained on such data [1] [6]. This divergence helps explain why strong laboratory results may not translate into operational effectiveness.

A second recurring bottleneck is labeling. Purely synthetic labels or weakly auditable procedures impair process reproducibility and ground-truth reliability. Assisted approaches that combine prior knowledge, log correlation (e.g., Suricata, Zeek), and targeted inspection tend to improve label quality and reproducibility [4] [6].

In recent years, several initiatives pursued both greater realism and better process documentation. CIC-IDS2017 and derivatives [1] expanded scenario coverage; TON_IOT increased source heterogeneity; and HIKARI-2021 explicitly emphasized encrypted traffic and the importance of documenting the creation pipeline, while also showing that background traffic may contain unplanned anomalies—a strong case for assisted labeling [3]. Reproducible attack-injection tools such as ID2T advocate replicating statistical properties of the background and embedding automated defect checks in the process [4] [7].

The recent literature also advances in domain-specific settings [8] [9] federated-learning scenarios [10], and temporal-drift evaluation [11] [12] in addition to systematic reviews with best-practice recommendations[13] and large-scale captures in real Internet environments[14].

Table I: Qualitative comparison with related work (2022–2025).

Reference	RC	LB	TMP	GEN	DF	MO
Edge-IIoT(2025)[8], [15]	✓	×	×	✓	✓	×
RT-IoT(2022)[9]	✓	×	×	✓	✓	×
FLNET(2023)[10]	✓	×	×	✓	✓	×
Koumar et al.(2023) [12]	✓	×	✓	✓	✓	✓
MAWIFlow(2025) [11]	✓	×	✓	✓	✓	✓
Wang et al.(2025)[14]	✓	×	✓	✓	✓	✓
Goldschmidt et al.(2025) [13]	✓	✓	✓	✓	✓	×
This Article	✓	✓	✓	✓	✓	✓

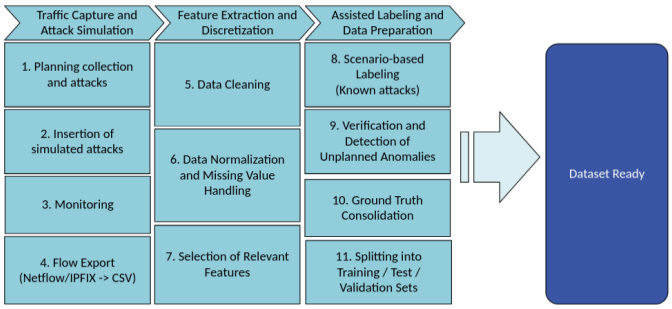

Figure 1: The three stages of the proposed methodology

Table I qualitatively summarizes these recent efforts along six objectives: realistic collection of benign traffic (RC), explicit and auditable labeling procedures (LB), temporally aware evaluation (TMP), evidence of generalization across network contexts (GEN), support for multiple data forms such as packet and flow records (DF), and a clearly stated model objective (MO). As the table shows, existing datasets and frameworks tend to satisfy only a subset of these criteria; none simultaneously combine realistic collection, assisted labeling, temporal validation, and generalization support with an explicit anomaly-detection objective. This motivates the unified pipeline proposed in this paper, which is designed to meet all six dimensions.

Despite these advances, a unified pipeline is still missing that simultaneously (i) continuously collects realistic benign traffic; (ii) injects controlled and reproducible attacks; (iii) produces assisted and auditable labels; (iv) validates temporally with distribution-stability metrics; and (v) publishes the process (scripts, configurations, and quality reports).

Finally, the *Energy-based Flow Classifier* (EFC) is a suitable case study: a one-class classifier that models benign traffic via inverse statistics (Potts model), offering interpretability and stronger robustness to distributional shifts [5].

III. Proposed Methodology

The proposed methodology covers two main and complementary fronts: (i) the generation of a realistic network traffic dataset with injected attacks for NIDS training and testing, and (ii) the application of the EFC algorithm on this dataset, including data preparation steps (attribute discretization) suitable for the model. The following description (Fig. 1) is organized into three stages: Traffic Capture and Attack Simulation, Feature Extraction and Discretization, and Assisted Labeling and Data Preparation.

1) Traffic capture and attack simulation: Benign traffic is collected in a real network or representative testbed while controlled attacks are executed (port scan, brute force, web exploits, DoS/DDoS, botnet). Traffic is mirrored at a central point and exported as NetFlow/IPFIX (flow metadata, low storage cost). IDS/monitors (Snort/Suricata, Zeek/Argus) assist validation. The capture should span different days and times (e.g., 5 days in CIC-IDS2017; 2 days in HIKARI-2021), acknowledging that not all remaining traffic is necessarily benign.
2) Feature extraction and discretization: With aggregated flow records, select relevant attributes (IPs, ports, protocol, bytes/packets, duration, direction, subnets) and apply discretization for EFC (bin 0 for zeros; quantiles for positive values). Perform basic cleaning (types, missing values, outliers) and, when useful, dimensionality reduction to simplify the model and speed up processing.
3) Assisted labeling and data preparation: Labels derive from scheduled attacks (IP/port/window) correlated with IDS logs. Remaining flows are checked with auxiliary detectors (offline IDS, outlier detection) to find unplanned anomalies; ambiguous cases may undergo manual review. Consolidate the ground truth (binary and optionally by attack type), reporting the natural imbalance. Finally, apply temporal splits (training on benign only; later validation/test with benign and malicious) to reflect prospective use and enable threshold/hyperparameter tuning.

IV. Case Study and Results

A. Experimental Setup

To validate the proposed methodology, we implemented a proof-of-concept architecture named "NetFlow Collector + Pipeline Dataset Construction for EFC". The environment was fully containerized with Docker, ensuring portability and reproducibility. The main components include:

- Collector service (NetFlow/IPFIX): generates raw flow files (nfcapd.*) and exports them to CSV format;
- Attacker orchestrator: containerized execution of predefined attack plans, injecting controlled malicious scenarios into the traffic;
- Pipeline scripts: process raw flows into discretized, EFC-ready datasets, with hashing-based anonymization and binning of continuous features (bytes, packets, duration);
- Labeling engine: aligns flow records with attack plans, enabling precise ground truth generation, both in anonymized and raw-IP modes;
- Validation modules: prepare temporal splits (T1/T2/T3) and check dataset stability with feature-distribution metrics;
- EFC runner: trains the Energy-based Flow Classifier on benign traffic and evaluates anomaly detection on subsequent temporal splits.

B. Dataset Characteristics

The pipeline produced a dataset combining real benign traffic with injected attack scenarios. Collection was carried out continuously over four consecutive days, capturing different time windows. The dataset includes:

- Volume: large-scale flow records exported in NetFlow/IPFIX format;
- Traffic diversity: cleartext and encrypted protocols (HTTP, HTTPS, DNS, email);
- Attack categories: port scans and TCP-SYN flood attacks;
- Ground truth: assisted labeling correlating attack plans, IDS/flow logs, and anomaly detection checks;
- Imbalance: as in real-world scenarios, benign flows largely outnumber malicious ones, providing a realistic ratio for anomaly detection research.

C. Evaluation with EFC

We evaluate the *Energy-based Flow Classifier* (EFC) under a temporal validation scheme with three non-overlapping windows: T1 (training, benign-only), and T2/T3 (validation/test with benign+malicious, unseen at training time). The EFC models benign traffic by inverse statistics (inverse Potts) over discretized features and assigns an "energy" to each flow; flows whose energy exceeds a calibrated threshold are flagged as anomalies.

a) Feature engineering (flows).: We discretize numeric attributes with a hybrid strategy (bin 0 for zeros; quantile bins for positive values) and categorical bucketization. In our real-traffic runs we used: $n\_bins = 30$,

Table II: EFC binary classification on real traffic (fixed cutoff quantile q=0.95).

Split	Class	Precision	Recall	F1	Support
T2	anomaly	0.74	0.94	0.83	10,841
	benign	0.99	0.95	0.97	71,225
Accuracy/FPR		0.95		FPR = 3520/71225 ≈ 4.9%	
T3	anomaly	0.62	0.95	0.75	14,267
	benign	0.99	0.92	0.95	101,592
Accuracy/FPR		0.92		FPR = 8436/101592 ≈ 8.3%	

port-topN = 50 for `svc_port_bucket`, and winsorization on `duration` at 1% (`-winsorize-duration-pct 0.01`). Direction (`lan2wan/wan2lan/intra/wan2wan`) and /24 subnets are included as categorical context.

b) Training and calibration.: We train EFC in one-class mode on T1 benign flows (`base_class=0`). Thresholds are calibrated from *benign energies* on downstream splits, using either a fixed quantile or a tuned cutoff.

c) Cutoff tuning.: Besides fixed quantiles, we employed an automatic tuner that scans quantiles q in a configurable range and chooses the cutoff by one of three objectives: (i) maximize F_1, (ii) maximize F_β (we used β=0.5 to prioritize precision), or (iii) minimize False Positive Rate (FPR) subject to a recall constraint ("constrained"). In practice we explored $q \in [0.90, 0.9995]$ (step 5×10^{-4}); when precision was a priority we restricted the search to the upper band, e.g., $q \in [0.985, 0.9995]$. Unless otherwise noted, the results below report a fixed cutoff quantile of 0.95 for comparability across T2 and T3.

d) Binary detection results.: Table II summarizes the binary (benign vs anomaly) performance on real traffic for T2 and T3 under the fixed quantile q=0.95. As expected, T3 shows higher FPR due to temporal drift, while recall remains high in both splits.

e) Energy histograms and confusion matrices.: Figures 2 and 3 (T2), and Figures 4 and 5 (T3) illustrate the separation of benign vs anomaly energies and the resulting confusion matrices at q=0.95. (Artifacts generated with our plotting utility; filenames indicated below.)

f) Notes on tuning.: When replacing the fixed quantile by the tuner, we used: (i) $q \in [0.90, 0.9995]$ (step 5×10^{-4}); (ii) F_β with β=0.5 to improve precision; and (iii) a constrained objective with target recall ≥ 0.90 to control FPR under drift (notably on T3). In practice we observed that restricting q to the upper band (≥ 0.985) substantially reduces FPR with marginal recall loss, which is desirable in operational NIDS settings.

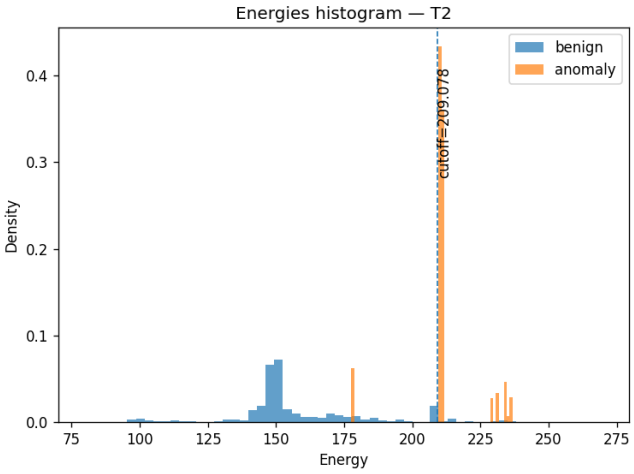

Figure 2: Energy histogram (T2).

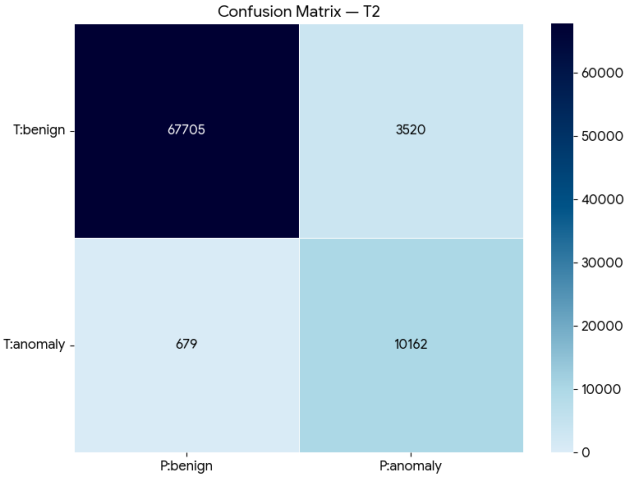

Figure 3: Confusion matrix (T2).

V. Discussion and Limitations

A. Summary

Our case study shows that the proposed pipeline produces realistic, auditable and temporally validated flow datasets, enabling effective training of anomaly-based models such as the Energy-based Flow Classifier (EFC). With a fixed cutoff quantile of q=0.95, the binary gate keeps high recall in both splits while precision degrades under drift: in T2 we obtain FPR $\approx 4.9\%$ and in T3 FPR $\approx 8.3\%$ with recall $\approx$ 0.94–0.95 (Table II, Figures 2–5).

B. Temporal drift.

The drift between T2 and T3 appears as a right-shift and larger overlap in the benign energy distribution, increasing false positives at the same operating point. To counter this, we adopted an automatic cutoff tuner that scans $q \in [0.90, 0.9995]$ (fine step) using: (i) F_1;

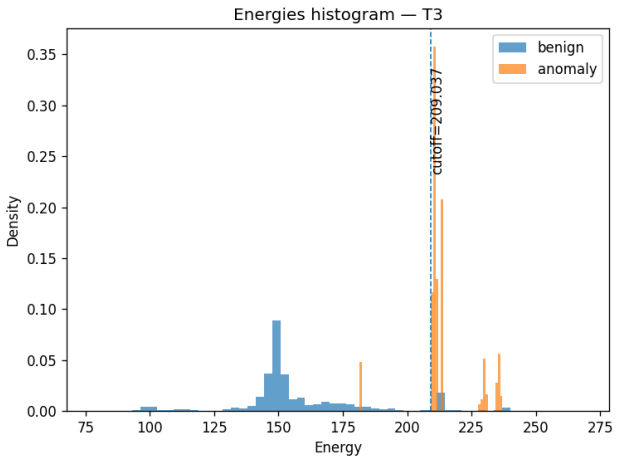

Figure 4: Energy histogram (T3).

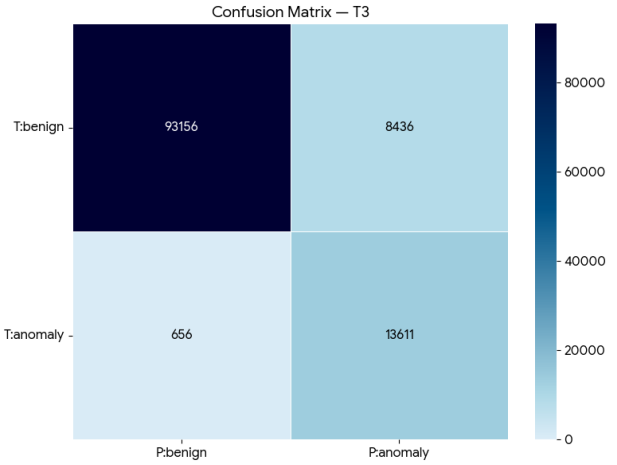

Figure 5: Confusion matrix (T3).

(ii) F_β with β=0.5 to prioritize precision; and (iii) a constrained objective that minimizes FPR subject to recall ≥ 0.90. Restricting the search to the upper band (e.g., $q \geq 0.985$) consistently reduced FPR in T3 with marginal recall loss, providing a reproducible way to set the operating point under drift.

C. Comparison with related work.

UNSW-NB15 [16] and CIC-IDS2017 [1] remain useful but suffer from dated patterns and ground-truth issues; HIKARI-2021 [17] and ID2T [7] advanced transparency and synthetic injection. Our contribution unifies these strengths with assisted labeling and *prospective* temporal splits, addressing reproducibility and enabling drift-aware evaluation—gaps still open in prior benchmarks.

D. Operational relevance.

Using NetFlow/IPFIX keeps the approach scalable and deployment-aligned; releasing anonymized flows together with scripts/configs/quality reports meets reproducibility and auditability goals highlighted by recent surveys [4] [13].

E. Limitations.

(1) *Attack coverage.* Current scenarios emphasize common classes (scan, brute-force, DoS/DDoS); stealthy and low-and-slow behaviors require expansion. (2) *Labeling scope.* Assisted labeling relies on predefined attack windows and IDS correlation; subtle out-of-window behaviors may remain unlabeled. (3) *Anonymization.* HMAC-based hashing preserves privacy but reduces IP-level interpretability in some analyses. (4) *Model scope.* While EFC handled drift reasonably, broader comparisons (e.g., federated or deep approaches [2]) are needed to assess generality. (5) *Binning sensitivity.* EFC depends on discretization (*n_bins*, service buckets); extreme settings may raise sparsity (precision loss) or over-smooth patterns (recall loss).

F. Future directions

Expand attack diversity (incl. encrypted-traffic anomalies), combine IDS-based labeling with active learning/visual validation, adopt group-specific cutoffs (e.g., per `proto/direction/svc_port`) with minimum benign support, and extend the pipeline to near-real-time with drift monitors that trigger re-tuning.

References

[1] S. Layeghy, M. Gallagher, and M. Portmann, "Benchmarking the benchmark — comparing synthetic and real-world network IDS datasets," *Journal of Information Security and Applications*, vol. 80, p. 103689, 2024.

[2] G. de Carvalho Bertoli, L. A. P. Junior, O. Saotome, and A. L. dos Santos, "Generalizing intrusion detection for heterogeneous networks: A stacked-unsupervised federated learning approach," *arXiv preprint arXiv:2209.00721*, 2022. [Online]. Available: https://arxiv.org/abs/2209.00721

[3] J. C. Mondragon, P. Branco, G. V. Jourdan, and A. A. Ghorbani, "Advanced ids: A comparative study of datasets and machine learning algorithms for network flow-based intrusion detection systems," *Applied Intelligence*, vol. 55, no. 1, p. 608, 2025. [Online]. Available: https://doi.org/10.1007/s10489-025-06422-4

[4] J. Guerra, C. Catania, and E. Veas, "Datasets are not enough: Challenges in labeling network traffic," *Comput. Secur.*, vol. 120, p. 102810, 2022.

[5] C. Pontes, M. Souza, J. Gondim, M. A. Marotta, and M. Bishop, "A new method for flow-based network intrusion detection using the inverse potts model," *IEEE Transactions on Network and Service Management*, vol. 18, no. 2, pp. 1125–1136, 2021.

[6] D. Pinto, I. Amorim, E. Maia, and I. Praça, "A review on intrusion detection datasets: tools, processes, and features," 5 2025.

[7] C. G. Cordero, E. Vasilomanolakis, A. Wainakh, M. Mühlhäuser, and S. Nadjm-Tehrani, "On generating network traffic datasets with synthetic attacks for intrusion detection," *ACM Transactions on Privacy and Security*, vol. 24, no. 2, pp. 8:1–8:39, 2020.

[8] M. A. Ferrag, O. Friha, D. Hamouda, L. Maglaras, and 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. 40 281–40 306, 2022.

[9] B. S. and R. Nagapadma, "RT-IoT2022 ," UCI Machine Learning Repository, 2023, DOI: https://doi.org/10.24432/C5P338.

[10] P. Kumar, J. Liu, A. S. M. Tayeen, S. Misra, H. Cao, J. Harikumar, and O. Perez, "Flnet2023: Realistic network intrusion detection dataset for federated learning," in *MILCOM 2023 - 2023 IEEE Military Communications Conference (MILCOM)*. IEEE, 10 2023, pp. 345–350.

[11] J. Schraven, A. Windmann, and O. Niggemann, "Mawiflow benchmark: Realistic flow-based evaluation for network intrusion detection," *arXiv preprint arXiv:2506.17041*, 6 2025. [Online]. Available: http://arxiv.org/abs/2506.17041

[12] J. Koumar, K. Hynek, and T. Čejka, "Network traffic classification based on single flow time series analysis," *arXiv preprint arXiv:2307.13434*, 7 2023. [Online]. Available: http://arxiv.org/abs/2307.13434

[13] P. Goldschmidt and D. Chudá, "Network intrusion datasets: A survey, limitations, and recommendations," *Computers & Security*, vol. 156, p. 104510, 2025. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0167404825001993

[14] W. Wang, W. Jiang, B. Zhang, Q. Zhu, and C. Liao, "A real network environment dataset for traffic analysis," *Scientific Data*, vol. 12, 12 2025.

[15] T. A. Nuaimi, S. A. Zaabi, M. Alyilieli, M. AlMaskari, S. Alblooshi, F. Alhabsi, M. F. B. Yusof, and A. A. Badawi, "A comparative evaluation of intrusion detection systems on the edge-iiot-2022 dataset," *Intelligent Systems with Applications*, vol. 20, 11 2023.

[16] N. Moustafa and J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in *Military Communications and Information Systems Conference (MilCIS)*. IEEE, 2015, pp. 1–6.

[17] A. Ferriyan, A. H. Thamrin, K. Takeda, and J. Murai, "Generating network intrusion detection dataset based on real and encrypted synthetic attack traffic," *Applied Sciences*, vol. 11, p. 7868, 2021.