Evaluating Lightweight Transformers for Network Intrusion Detection under Temporal Drift: A Comparative Study with MAWIFlow

João Jardim Neto[1], **César Augusto Cavalheiro Marcondes**[1]

[1]Computer Science Division – Aeronautics Institute of Technology (ITA)
12.228-900 – São José dos Campos – SP – Brazil

joao.neto@ga.ita.br, cesar.marcondes@gp.ita.br

Abstract. *Static train/test splits remain the dominant evaluation protocol for network intrusion detection systems (NIDS), yet real-world traffic evolves continuously and deployed models degrade under temporal drift. We present a reproducible comparative evaluation of four model families—XGBoost, MLP, CNN-BiLSTM, and a parameter-efficient causal Transformer (LightTransformer, a compact FlowTransformer-derived configuration)—on two complementary protocols: a static split on CICIoT2023 (sanity check) and a temporal forward-chaining evaluation on MAWIFlow (2007–2024). On the static benchmark all models achieve $F1_+ > 0.98$ on the attack class, so near-ceiling static scores are insufficient evidence of temporal robustness. Under temporal drift (cumulative training window, $R=5$ seeds, all models) the LightTransformer obtains $nAUT_1 = 0.57$ (95% CI [0.52, 0.62]) versus 0.44 (CNN-BiLSTM), 0.38 (XGBoost), and 0.28 (MLP). Bootstrap confidence intervals are consistent with this ordering, which holds across all five seeds; a Friedman test ($\chi^2 = 15.0$, $p = 0.002$) rejects equal rankings globally; one-sided pairwise Wilcoxon signed-rank tests provide uncorrected directional evidence ($p = 0.031$ for all six pairs, $n = 5$); and no pairwise comparison survives Bonferroni correction ($\alpha_c = 0.0083$). Our results indicate that forward-chaining temporal evaluation should be a standard component of deployment-oriented NIDS benchmarking.*

1. Introduction

Network intrusion detection systems (NIDS) form a critical layer of modern cybersecurity infrastructure, monitoring network flows to identify malicious activity in real time. The success of machine learning (ML) in classification tasks has driven extensive research on automated, feature-based NIDS, with methods ranging from gradient-boosted trees [Chen and Guestrin 2016] to diverse deep learning architectures [Lansky et al. 2021, Chinnasamy et al. 2025]. Systematic surveys document the rapid growth of Transformer-oriented IDS and identify scalability, dataset dependence, and temporal adaptability as open challenges [Çağdaş Özer and Orman 2024, Kheddar 2025]. Manocchio et al. [Manocchio et al. 2024] systematically evaluated multiple Transformer components for flow-based NIDS and identified compact configurations that achieve competitive performance with reduced model size.

Despite this progress, the dominant evaluation methodology suffers from a fundamental limitation: the use of static, time-agnostic train/test splits. In practice, network traffic distributions shift continuously due to the emergence of new attack campaigns, changes in legitimate user behavior, and infrastructure evolution, a

phenomenon known as *concept drift* [Lu et al. 2018] (referred to as *temporal drift* throughout this paper). A model with near-perfect F1 on a test set drawn from its own collection period may degrade drastically when deployed months or years later. Sommer and Paxson [Sommer and Paxson 2010] highlighted fundamental evaluation and deployment challenges for ML-based NIDS; more recent dataset surveys [Goldschmidt and Chudá 2025] document timeliness and split quality as persistent dataset limitations, and the MAWIFlow benchmark [Schraven et al. 2026] operationalizes this concern with a temporal forward-chaining protocol.

The MAWIFlow dataset and benchmark [Schraven et al. 2026] provide a principled framework for temporal evaluation. Derived from a trans-Pacific backbone link, MAWIFlow spans 18 annual snapshots (2007–2024) with binary anomaly labels derived from MAWILab v1.1 anomaly annotations [Fontugne et al. 2010]. Its forward-chaining evaluation protocol trains on historical data and tests exclusively on future windows, directly measuring a model's ability to generalize under temporal drift. The benchmark defines a horizon-limited normalized Area Under Time (nAUT) metric, here denoted nAUT_H for horizon H, which summarizes model performance as a function of temporal distance from the training window.

In this paper we evaluate four representative model families under both a standard static protocol and the MAWIFlow forward-chaining protocol: (1) **XGBoost** [Chen and Guestrin 2016], a scalable gradient-boosted tree ensemble and strong tabular baseline; (2) **MLP**, a neural network baseline for tabular data; (3) **CNN-BiLSTM**, a sequential recurrent model operating on windows of consecutive flows; and (4) **LightTransformer** (**LT**), implementing a compact FlowTransformer-derived configuration based on [Manocchio et al. 2024] in PyTorch.

This study is guided by two research questions and one empirical hypothesis. RQ_1*:* Does near-ceiling performance under a conventional static NIDS benchmark (CICIoT2023 [Neto et al. 2023]) constitute sufficient evidence of robustness under temporal drift? RQ_2*:* Does a parameter-efficient causal Transformer retain higher temporal performance than tabular and recurrent baselines under forward-chaining evaluation? H_1*:* Sequential flow representations, modeled as windows of $W \times F$ consecutive flow records, achieve higher temporal robustness (measured by nAUT_H) than tabular $1 \times F$ representations under temporal drift; H_1 is evaluated here under the cumulative training-window forward-chaining protocol (Section 3). Our contributions are:

- A static four-way comparison on CICIoT2023 ($R{=}5$ seeds; Table 5), serving as a complementary static reference and sanity check for RQ_1: all four families achieve competitive attack-class F1, yet near-ceiling scores are insufficient evidence of temporal robustness, motivating the MAWIFlow evaluation.
- To the best of our knowledge, this is the first evaluation of a compact FlowTransformer-derived configuration based on [Manocchio et al. 2024] under MAWIFlow temporal forward-chaining (18 annual snapshots, 2007–2024), directly addressing RQ_2 and testing H_1: under the cumulative training window, LightTransformer leads all models ($\text{nAUT}_1{=}0.57$; full ordering LT $>$ CNN-BiLSTM $>$ XGBoost $>$ MLP, consistent across all $R{=}5$ seeds; statistical evidence in Section 4); causal attribution requires future ablation.
- A reproducible experimental infrastructure: Optuna HPO [Akiba et al. 2019],

fixed random seeds, JSON result logs, and a PyTorch reimplementation of the FlowTransformer architecture [Manocchio et al. 2024]; HPO and reproducibility details in Tables 3 and 4.

2. Related Work

2.1. Deep Learning for Network Intrusion Detection

Systematic reviews document the maturity and architectural diversity of DL-based NIDS. Chinnasamy et al. [Chinnasamy et al. 2025] present a PRISMA-based systematic review, cataloging architectures from autoencoders and DNNs to CNNs, RNNs, and LSTMs [Lansky et al. 2021], and identifying open challenges in generalization, class imbalance, and dataset quality. A complementary survey [Abdulganiyu et al. 2023] corroborates that the field has expanded substantially but lacks consensus on evaluation protocols for deployment. At the architecture level, CNNs have been applied to end-to-end encrypted traffic classification from raw session and flow bytes [Wang et al. 2017]; RNN-based IDS models have been evaluated on static connection-level benchmarks such as NSL-KDD [Yin et al. 2017].

2.2. Transformers in Intrusion Detection Systems

Transformer-based architectures have emerged as a growing research line within IDS. A systematic mapping study identified 697 publications on the topic, documenting an evolution from early explorations of attention mechanisms to specialized applications in IoT security, real-time detection, and hybrid CNN-LSTM Transformer designs [Çağdaş Özer and Orman 2024]. A complementary survey of Transformer and LLM variants corroborates this trend, highlighting open challenges in scalability, dataset dependence, and temporal adaptability [Kheddar 2025]. Wu et al. [Wu et al. 2022] proposed RTIDS, a stacked encoder-decoder Transformer with positional embedding and self-attention, evaluated on CICIDS2017 and CIC-DDoS2019; Decision Transformer-based approaches have also been applied to sequential packet-level detection in an offline reinforcement-learning setting [Zhou et al. 2024].

Among these, Manocchio et al. [Manocchio et al. 2024] conducted a systematic ablation of Transformer configurations for NIDS, varying layers, attention heads, input encoding strategies, and classification heads. Based on component-level findings in that study, we adopt a *decoder-only causal Transformer* with 2 layers, 2 attention heads, $d_{\text{model}}{=}128$, record-level projection as input encoding, and a Last Token classification head, a configuration that achieves competitive accuracy with substantially fewer parameters than encoder-based alternatives. We reimplement this configuration in PyTorch and extend its evaluation to the temporal forward-chaining protocol.

2.3. MAWIFlow Temporal Benchmark

Schraven et al. [Schraven et al. 2026] introduce MAWIFlow, a NIDS benchmark derived from MAWI backbone traffic, together with a temporal forward-chaining evaluation protocol designed to measure robustness to temporal drift. Their protocol evaluates models across multiple training-window variants $k{\in}\{1, 2, 3, \text{cumulative}\}$ and reports horizon-limited nAUT_H metrics. The original benchmark uses annual forward-chaining evaluation with a binary task (benign vs. anomalous); taxonomy-wise analyses (12 higher-level

MAWILab groups) are a complementary component [Schraven et al. 2026]. Our work builds on the MAWIFlow dataset and forward-chaining protocol, adding the LightTransformer causal decoder configuration absent from the original model set; our evaluation setting differs from [Schraven et al. 2026] (notably the native, imbalanced class distribution of the sampled days rather than the original class-balanced sampled subset), so absolute scores are not directly comparable (see Section 4). We further complement the temporal axis with a static evaluation on CICIoT2023.

2.4. CICIoT2023

The CICIoT2023 dataset [Neto et al. 2023] captures traffic from 105 IoT devices under 33 distinct attack types organized into 7 categories. With approximately 46.7 million labeled flows and 39 numeric features, it provides a large-scale static benchmark suitable for evaluating model capacity on a class-imbalanced problem. We use CICIoT2023 as a *sanity-check* static axis, ensuring all four model families are competitive before the temporal evaluation; the inflation risk of non-time-respecting splits [Goldschmidt and Chudá 2025] appears here as near-ceiling attack-class F1 scores.

Research gap. Table 1 positions the six most directly related empirical works against the present study. Two complementary gaps emerge from this comparison. First, the compact FlowTransformer configuration identified in [Manocchio et al. 2024] has been validated only on static benchmarks; its behavior under temporal concept drift is unknown. Second, the MAWIFlow temporal benchmark [Schraven et al. 2026] evaluates drift robustness across multiple model families under annual forward-chaining evaluation and binary classification, but includes no Transformer baseline, leaving the temporal robustness of a lightweight causal decoder architecture unexamined within this evaluation paradigm. Sequential and convolutional baselines [Yin et al. 2017, Wu et al. 2022, Wang et al. 2017] also employ static evaluation, and the systematic reviews [Chinnasamy et al. 2025, Kheddar 2025, Çağdaş Özer and Orman 2024, Abdulganiyu et al. 2023], which aggregate empirical works and therefore do not appear as table rows, corroborate these gaps at the meta-level. The present paper addresses both gaps simultaneously: it applies a compact FlowTransformer-derived configuration based on [Manocchio et al. 2024] to the MAWIFlow forward-chaining protocol, enabling, to the best of our knowledge, the first comparison of LightTransformer, CNN-BiLSTM, XGBoost, and MLP under annual temporal drift with binary labels on a real-world backbone dataset.

3. Methodology

3.1. Datasets

CICIoT2023. We use a consolidated Parquet version of CICIoT2023 [Neto et al. 2023] containing 46.7 million flows and 39 numeric features (consolidation, binary label derivation from the source CSV subdirectory, and the 39-feature schema are products of our preprocessing pipeline). Labels are binary (0 = benign, 1 = attack), with severe class imbalance (97.7% attack). Neural models use inverse-frequency class weights in the cross-entropy loss ($w_c = N/(K \cdot N_c)$, derived from training labels). For XGBoost, `scale_pos_weight` is treated as an Optuna hyperparameter (range $[1, 50]$); the selected value (≈ 1.4) is near-neutral, reflecting that label 1 is already the majority class.

Table 1. Positioning of related empirical work.

Ref.	Dataset	Models	Fwd. chain	nAUT metric	Trans-former	Real backbone
W1	NSL-KDD, UNSW-NB15, CSE-CIC-IDS2018	FlowTransformer (enc./dec.; GPT/BERT; ablation)	✗	✗	✓	✗
W2	MAWIFlow (MAWILab v1.1, 2007–2024)	CNN-BiLSTM, DT, IF, LR, RF, XGBoost	✓	✓	✗	✓
W3	CICIDS2017, CIC-DDoS2019	RTIDS: encoder-decoder Transformer	✗	✗	✓	✗
W4	UNSW-NB15 (packet-level)	Decision Transformer (offline RL)	✗	✗	✓	✗
W5	NSL-KDD	RNN-IDS	✗	✗	✗	✗
W6	ISCX VPN-nonVPN	End-to-end 1D-CNN	✗	✗	✗	✗
This paper	**MAWIFlow + CICIoT2023**	**XGBoost, MLP, CNN-BiLSTM, LightTrans-former**	✓	✓	✓	✓

Notes: ✓ = present; ✗ = absent or not applicable. W1 [Manocchio et al. 2024]; W2 [Schraven et al. 2026]; W3 [Wu et al. 2022]; W4 [Zhou et al. 2024]; W5 [Yin et al. 2017]; W6 [Wang et al. 2017].

We apply a seeded (`random_state=42`), label-stratified static 60/20/20 split, yielding approximately 28.0M training, 9.3M validation, and 9.3M test samples; stratification preserves the 97.7 %/2.3 % class balance across all three partitions. Because the CICIoT2023 representation used in this work does not provide a real timestamp column, this split is not interpreted as temporal; it serves only as a static sanity-check axis. Log-transform and min-max normalization are applied to all numeric features using statistics computed on the training split only.

MAWIFlow. We use the binary-labeled MAWIFlow dataset spanning annual snapshots from 2007 to 2024 (18 time points), derived from trans-Pacific MAWI backbone traffic. Following the public MAWIFlow sampling layout, each annual snapshot is assembled from a few systematically sampled capture days (one per quarter where available; 74 days and $\approx$9.9M flows in total across 2007–2024); we therefore evaluate on a MAWIFlow *subset*. Within each sampled day all flows are retained, preserving the native, unbalanced class distribution rather than class-resampling; anomaly prevalence varies widely across years (2.1% in 2017 to 86.3% in 2023). Per-year capture days, flow counts and class balance are tabulated in `results/paper_artifacts/data_manifest.md`. After removing metadata columns (label, year, timestamp), our preprocessing pipeline re-

tains 89 numeric features for model input. No additional resampling is applied by our pipeline; class imbalance is addressed through per-class weighting at training time rather than sampling. The same log-transform and per-window min-max normalization pipeline is used. The forward-chaining protocol recomputes normalization statistics from each training window to prevent data leakage.

3.2. Model Families

We evaluate four model families, summarized in Table 2. Tabular models (XGBoost, MLP) receive individual flow feature vectors; sequential models (CNN-BiLSTM, LightTransformer) receive fixed-size windows of $W{=}16$ consecutive flow vectors, constructed via a sliding window over each data split. The window length $W{=}16$ is treated as a fixed, representative value rather than a tuned hyperparameter; a systematic window-size ablation is left to future work.

Table 2. Model families evaluated. W = window size, F = feature dimension.

Model	Type	Input	Framework
XGBoost	Tabular ensemble	$1{\times}F$	xgboost
MLP	Tabular neural	$1{\times}F$	PyTorch
CNN-BiLSTM	Sequential recurrent	$W{\times}F$	PyTorch
LightTransformer	Causal decoder	$W{\times}F$	PyTorch

Sequence construction. For sequential models, flows within each split (training window or individual test-year snapshot) are sorted chronologically by timestamp prior to windowing, so that position i in every window temporally precedes position $i{+}1$; this ordering is a prerequisite for the causal mask to encode genuine temporal precedence. The timestamp column is then excluded as a feature before the preprocessing pipeline is applied. A sliding window of size $W{=}16$ is applied with stride $s{=}W/2{=}8$ during training (50 % overlap) and stride $s{=}1$ during evaluation (one window per flow record). The first $W{-}1$ positions of each split are left-zero-padded (every flow anchors one evaluation window), and the window label is that of the *final* flow, consistent with the Last Token head of both architectures. Windows never cross split boundaries: the training set and each annual test snapshot are windowed independently, preventing label leakage across temporal splits. Within a multi-year training window, the chronologically sorted flows form one continuous sequence, so training windows may span adjacent years; each annual test snapshot is instead windowed from a cold start. On Axis 1, CICIoT2023 provides no timestamp, so windows are built over the rows of each stratified split in their seeded random order using the same machinery (stride, padding, last-token label); these windows give the sequential architectures a fixed-size $W{\times}F$ input but carry no temporal ordering, and Axis-1 results for the sequential models should be read as static discriminative performance, not as temporal sequence modeling.

The **LightTransformer** architecture is fixed to a compact configuration based on the FlowTransformer study [Manocchio et al. 2024]: a 2-layer causal decoder Transformer with 2 attention heads, $d_{\text{model}}{=}128$, $d_{\text{ff}}{=}512$, sinusoidal positional encoding, record-level projection as input encoding (a linear layer mapping each flow vector to d_{model}

before position-wise attention), and a Last Token classification head that reads the final position's representation for binary prediction. This configuration is not subject to architecture search, only training hyperparameters (learning rate, weight decay, dropout) are optimized. The model has approximately 416K trainable parameters on the MAWIFlow axis (F=89 features; ≈410K on the CICIoT2023 axis, F=39), of which only 11K are feature-dependent (the input projection); the remaining 405K reside in the fixed decoder layers and classification head. This is aligned with the [Manocchio et al. 2024] finding that appropriate input encoding and classification head choices can substantially reduce model size without sacrificing accuracy.

The **CNN-BiLSTM** applies one or two stacked 1D convolutional blocks (kernel size 3, BatchNorm1d, ReLU) for local feature extraction, followed by a single-layer bidirectional LSTM and a linear head applied to the *last timestep* of the LSTM output. Convolutional channel widths and LSTM hidden size are part of the HPO search.

The **MLP** applies a sequence of Linear → BatchNorm1d → ReLU → Dropout blocks, one per hidden layer, followed by a linear output. Layer count and hidden widths are drawn from five preset configurations searched by HPO.

3.3. Hyperparameter Optimization

We use Optuna [Akiba et al. 2019] with the TPE sampler for hyperparameter optimization (HPO) on each model. The number of Optuna trials varies by model due to per-trial computational cost (Table 3); these decisions are documented as Architecture Decision Records (ADRs) in the project repository. HPO budgets for CNN-BiLSTM and LightTransformer are intentionally constrained by per-trial computational cost and should not be interpreted as exhaustive searches; the MAWIFlow HPO-once policy is described under Evaluation Protocols.

Table 3. Budget-limited hyperparameter search per model.

Model	Trials (Ax. 1)	Trials (Ax. 2)	Cost/trial	Rationale
XGBoost	50	50	≈1 min	Full budget
MLP	15	5	≈44 min	Reduced budget
CNN-BiLSTM	2	5	≈112 min	Reduced budget
LightTransformer	1	5	≈154 min	Cost-bound; arch. fixed

Per-trial cost measured on Axis 1 (CICIoT2023).

The HPO objective in all cases is **validation F1$_+$** (positive-class F1 on the attack label), used uniformly across both axes for methodological consistency: on Axis 2 the central metric nAUT_H is computed from the future-year F1$_+$ trajectory (Eq. 1), so the HPO objective is aligned with, but does not directly optimize, the temporal robustness metric (metric-reporting choices are discussed in the Metrics subsection). For PyTorch models, each trial trains for up to 30 epochs; per-trial early stopping monitors validation F1$_+$ with a patience of 5 epochs and restores the best checkpoint. Unpromising trials are pruned by Optuna's **MedianPruner** (n_{startup}=5 trials, n_{warmup}=5 steps). Complete search space definitions are in `src/lift_nids/training/optuna_search.py` (Table 3); per-trial results and final hyperparameter values are logged in per-model `optuna.db` files under `data/results/axis{1,2}_final/<model>/`.

3.4. Evaluation Protocols

Scope: no cross-dataset transfer. Cross-dataset generalization protocols train a model on a source dataset and evaluate it, without retraining, on a distinct target dataset, quantifying transfer across network environments [Apruzzese et al. 2022]. This study performs no such transfer: each model is trained and evaluated independently within each axis, and no model crosses datasets. CICIoT2023 and MAWIFlow are complementary benchmarks addressing different evaluation questions, static discriminative capacity and temporal robustness, respectively; cross-dataset transfer between them remains future work.

Axis 1 — Static (CICIoT2023). Axis 1 addresses RQ_1 as a complementary static reference and sanity check: competitive performance here is necessary but not sufficient evidence for temporal robustness. Models are trained on the 60% split, validated on 20%, and evaluated on the held-out 20% test set. Each final model is retrained $R{=}5$ times with different random seeds ({42, 123, 456, 789, 1024}). Reported metrics are means across seeds with bootstrap 95% confidence intervals ($B{=}1000$).

Axis 2 — Temporal Forward-Chaining (MAWIFlow). Axis 2 is the primary inferential axis: it addresses RQ_2 and provides empirical evidence on H_1 by measuring whether sequential models retain higher $\mathrm{F1}_+$ under temporal drift. MAWIFlow is the principal dataset for all conclusions about temporal robustness; CICIoT2023 serves as a complementary static reference only.

Let $\mathcal{D}_y$ denote the annual snapshot for year y. We implement four training-window variants: $k{\in}\{1, 2, 3, \mathrm{cumulative}\}$. For $k{\in}\{1, 2, 3\}$, each sliding training window covers k consecutive years, $\mathcal{W}_i^{(k)} = \{y_i, \ldots, y_{i+k-1}\}$; in the cumulative variant the window grows monotonically, $\mathcal{W}_i^{(\mathrm{cum})} = \{y_0, \ldots, y_i\}$ (where $y_0{=}2007$). In every variant the *anchor* $a_i = \max \mathcal{W}_i$ is the most recent training year, the forecast origin from which the future is measured: snapshots $\mathcal{D}_y$ with $y{>}a_i$ serve as test sets at lag $\Delta t = y - a_i$. The cumulative variant is the primary comparison axis because it approximates an operational scenario in which a model is periodically retrained on all available historical data. Fixed-window variants ($k{\in}\{1, 2, 3\}$) were also executed; they are summarized descriptively in Section 4, where the model ranking is *not* invariant to the training-window regime. Accordingly, all comparative conclusions in this paper are scoped to the cumulative training-window strategy.

Hyperparameters are optimized once on $\mathcal{D}_{2007}$ via budget-limited Optuna search (TPE sampler; per-model budget N in Table 3), using a temporal 60/20/20 split into fit, validation, and internal-test subsets; the internal-test subset is held out during HPO. The resulting configuration θ^* is reused across all subsequent training windows and seeds (HPO-once policy). For each window $\mathcal{W}_i^{(k)}$ and seed s, a model is trained on the 60% temporal fit split of $\bigcup_{y \in \mathcal{W}_i^{(k)}} \mathcal{D}_y$ and evaluated on that window's own internal-test subset at lag $\Delta t{=}0$ and on each future snapshot $\mathcal{D}_y$, $y{>}a_i$, at lag $\Delta t = y - a_i$. Per-seed median $\mathrm{F1}_+$ trajectories across windows are summarized by the trapezoidal rule (Eq. 1), yielding $\mathrm{nAUT}_H^{(s)}$ averaged over seeds as $\overline{\mathrm{nAUT}}_H$; Figure 1 provides a visual overview. *All four model families (XGBoost, MLP, CNN-BiLSTM, LightTransformer) use full* $R{=}5$ *production runs with seeds* $\{42, 123, 456, 789, 1024\}$.

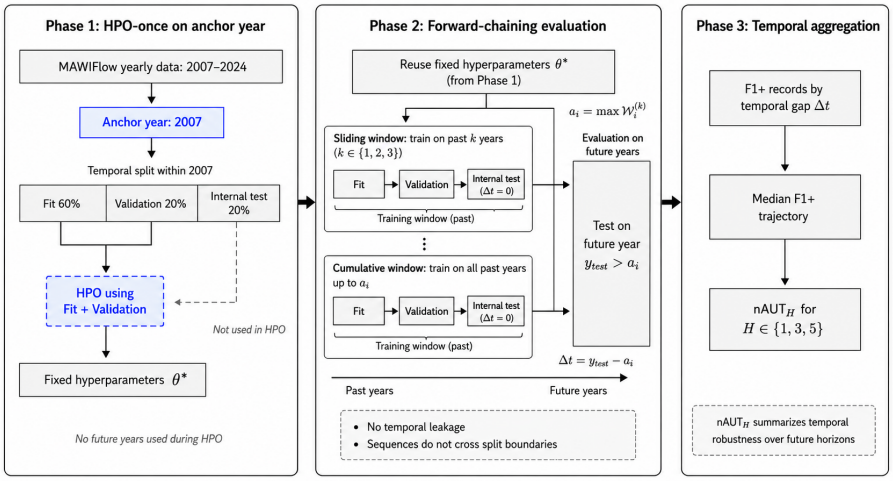

Figure 1. Axis 2 forward-chaining evaluation with HPO-once on MAWIFlow.

3.5. Metrics

Classification metrics. We report macro-averaged F1 ($\mathrm{F1}_{\mathrm{macro}}$), F1 on the attack class ($\mathrm{F1}_{+}$), and, on Axis 1, area under the ROC curve (AUC-ROC). AUC-PR is not reported: on CICIoT2023 the attack class is the *majority* (97.7%), so AUC-PR for attacks would be near-ceiling and uninformative, while per-class AUC-PR for benign flows was not computed. $\mathrm{F1}_{\mathrm{macro}}$—which equally weights both classes—is the primary ranking metric for Axis 1, as it is more discriminative than $\mathrm{F1}_{+}$ under this severe class imbalance. AUC-ROC is retained only for comparability with prior work, where it is conventionally reported; it is near-ceiling for all models (0.995–0.999, Table 5) and, like $\mathrm{F1}_{+}$, is not discriminative under this imbalance.

Temporal robustness metric. Following [Schraven et al. 2026], we define the horizon-limited normalized Area Under Time metric nAUT. Recall that $a_i = \max \mathcal{W}_i$ is the anchor of training window i, and let $\Delta t = y - a_i \geq 0$ be the lag to evaluation year $y \geq a_i$; $\Delta t{=}0$ is realized by the window's internal-test subset, as $\mathcal{D}_{a_i}$ is itself partly training data (see Figure 1). For each seed s, let $\widetilde{\mathrm{F1}}_{+}^{(s)}(\Delta t)$ denote the median attack-class F1 over all window positions i that yield lag Δt. We compute nAUT per seed via the trapezoidal rule over $\Delta t \in [0, H]$:

$$\mathrm{nAUT}_H^{(s)} = \frac{1}{H} \sum_{\Delta t=0}^{H-1} \frac{\widetilde{\mathrm{F1}}_{+}^{(s)}(\Delta t) + \widetilde{\mathrm{F1}}_{+}^{(s)}(\Delta t{+}1)}{2}, \tag{1}$$

and report the mean over seeds $\mathcal{S} = \{42, 123, 456, 789, 1024\}$: $\overline{\mathrm{nAUT}}_H = \frac{1}{|\mathcal{S}|} \sum_{s \in \mathcal{S}} \mathrm{nAUT}_H^{(s)}$. We report $\overline{\mathrm{nAUT}}_H$ for $H \in \{1, 3, 5\}$; a higher value indicates the model retains predictive accuracy further into the future under temporal drift. Because $\Delta t{=}0$ is the internal-test subset, nAUT_1 weights in-distribution and one-year-ahead performance equally; the two components are not separately identifiable from the aggregate score.

3.6. Reproducibility and Artifact Availability

Table 4 consolidates the software environment, random seeds, entry-point scripts, dataset dimensions, and approximate runtimes for both experimental axes. Hyperparameter optimization trial budgets and per-trial cost rationale appear in Table 3; the HPO objective is validation $F1_+$ in all cases. Source code, configuration files, per-seed JSON result artifacts, and statistical-test outputs are publicly available in the artifact repository under an immutable release tag.[1]

Table 4. Key reproducibility parameters for both experimental axes.

Item	Value
Axis 1 — CICIoT2023 (static); entry script `scripts/run_axis1.py`	
Dataset / partitions	46.7M flows; 39 features; 28.0M / 9.3M / 9.3M (60/20/20).
Approx. runtime	XGB ≈1.0 h; MLP ≈2.5 h; CNN-BiLSTM ≈14.7 h; LT ≈5.1 h.
Axis 2 — MAWIFlow (temporal); entry script `scripts/run_axis2.py`	
Dataset	18 annual snapshots (2007–2024); 89 features.
Approx. runtime	XGB ≈4.6 h; MLP ≈15.5 h; CNN-BiLSTM ≈16.6 h; LT ≈28.6 h (R=5, all k).

Note: Both axes use R=5 seeds $\{42, 123, 456, 789, 1024\}$ and the TPE sampler (Table 3). Environment: Python ≥3.11, PyTorch ≥2.1 (CUDA), XGBoost ≥2.0, scikit-learn ≥1.4, Optuna ≥3.6; pinned versions in the project lock file. Hardware: Intel i7-11800H, 32 GB RAM, RTX 3070 Laptop (8 GB VRAM).

4. Experimental Results

4.1. Axis 1: Static Evaluation on CICIoT2023

Table 5 reports performance on the static CICIoT2023 test set across R=5 seeds. All models achieve $F1_+ > 0.98$, reflecting the severe class imbalance (97.7% attack) that makes the attack-class F1 an almost trivially achievable threshold on this benchmark. Macro-averaged F1 is the more discriminative metric: XGBoost achieves the highest $F1_{macro}$ (0.928), while the neural models trail by 6–9 percentage points. This ranking is on a metric that was not the HPO objective (validation $F1_+$; Section 3), so the neural models may be sub-optimally calibrated for the benign class (a construct-validity caveat discussed in Section 5). The HPO budget asymmetry on this axis also runs in XGBoost's favor (50 trials vs. 1–15 for the neural models; Table 3), so the macro-F1 gap should be read with both factors in mind. These results address RQ_1: near-ceiling $F1_+$ scores reveal a saturation effect, making the CICIoT2023 static axis insufficient as evidence of temporal robustness. All four families are competitive (sanity check); the primary robustness comparison is provided by the temporal evaluation in Axis 2, where alignment between the HPO objective ($F1_+$) and the reported metric ($nAUT_H$) is direct.

The narrow cross-seed standard deviations (≤ 0.01 for $F1_{macro}$ across all models) reflect a low-variance experimental setup: the CICIoT2023 static split is highly reproducible. LightTransformer exhibits the largest cross-seed variance (std=0.009 for $F1_{macro}$),

[1] `https://github.com/JohnGarden/sbseg26-LiFT-NIDS` — release `v1.0.0-sbseg2026-camera-ready`.

Table 5. Axis 1: CICIoT2023 test-set performance (R=5 seeds, means).

Model	$\mathbf{F1}_{\text{macro}}$	$\mathbf{F1}_{+}$	AUC-ROC
XGBoost	**0.928**	**0.997**	**0.999**
MLP	0.864	0.991	0.998
CNN-BiLSTM	0.840	0.989	0.996
LightTransformer	0.841	0.989	0.995

followed by CNN-BiLSTM (std=0.004); these are also the two models with the most constrained Axis-1 HPO budgets (1 and 2 trials, respectively; Table 3), so a broader search could yield more stable optima.

An important caveat is that all four models struggle on the minority class (benign; 2.3% of flows): benign-class F1 (from $\mathrm{F1}_0 = 2\,\mathrm{F1}_{\text{macro}} - \mathrm{F1}_{+}$, not tabulated) ranges from $\approx$0.69 (CNN-BiLSTM) to 0.86 (XGBoost), consistent with class-weight correction improving but not fully resolving the imbalance despite high $\mathrm{F1}_{+}$.

4.2. Axis 2: Temporal Evaluation on MAWIFlow

Table 6 reports nAUT_H scores for the cumulative training-window variant on MAWIFlow, addressing RQ_2 and H_1. All four models use full production evaluation (R=5 seeds).

Table 6. MAWIFlow nAUT_H results for Axis 2 (cumulative window, R=5; means with bootstrap 95% CI [lower, upper]).

Model	$\mathbf{nAUT}_1$	$\mathbf{nAUT}_3$	$\mathbf{nAUT}_5$
XGBoost	0.377 [0.350, 0.396]	0.325 [0.308, 0.337]	0.285 [0.269, 0.297]
MLP	0.282 [0.260, 0.303]	0.210 [0.203, 0.221]	0.165 [0.158, 0.173]
CNN-BiLSTM	0.439 [0.418, 0.463]	0.358 [0.338, 0.377]	0.289 [0.273, 0.303]
LightTransformer	**0.568** [0.519, 0.618]	**0.474** [0.437, 0.503]	**0.386** [0.360, 0.409]

The performance ordering under temporal evaluation differs markedly from Axis 1. LightTransformer achieves nAUT_1=0.568, approximately 29% higher than CNN-BiLSTM (0.439) and more than twice the MLP (0.282); CNN-BiLSTM in turn exceeds XGBoost (0.377), with bootstrap CIs consistent with this ordering (Table 6). The MLP, with no capacity to exploit sequential structure, shows the lowest nAUT at every horizon. The ranking LT > CNN-BiLSTM > XGBoost > MLP holds for all five seeds at nAUT_1 and nAUT_3; at nAUT_5 the CNN-BiLSTM vs. XGBoost sub-ordering reverses in 2 of 5 seeds (consistent with the overlapping CIs at H=5). The statistical evidence follows a fixed hierarchy. First, a Friedman test (χ^2=15.0, p=0.002; the maximum achievable statistic for k=4 groups and n=5 blocks, i.e. perfect rank consistency on nAUT_1) rejects equal rankings globally. Second, one-sided pairwise Wilcoxon signed-rank tests on nAUT_1 yield p=0.031 for all six pairs (the minimum attainable at n=5; all paired differences favour the same model in every seed), providing uncorrected directional evidence. Third, no pairwise test remains significant after Bonferroni correction for six comparisons (α_c=0.05/6$\approx$0.0083), so we claim no corrected pairwise significance. A Nemenyi

post-hoc on the same ranks ($\mathrm{CD}{=}2.10$; $k{=}4$, $n{=}5$) separates only LightTransformer from MLP; all other pairs fall within the critical difference, consistent with the Bonferroni outcome. The bootstrap 95% CIs (Table 6) are consistent with the observed ordering—the LightTransformer and CNN-BiLSTM intervals do not overlap at any horizon, while CNN-BiLSTM vs. XGBoost overlap at the longest horizon ($H{=}5$), but interval overlap is not interpreted as a formal paired test. The LightTransformer exhibits the largest cross-seed variance (nAUT_1 std=0.063, range 0.498–0.639), reflecting initialization sensitivity; its ordering advantage over CNN-BiLSTM nonetheless holds in all five seeds without exception.

The degradation pattern across horizons ($H{=}1{\to}3{\to}5$) is consistent for all models: longer forecasting horizons show lower nAUT, as expected. The absolute gap between LT and CNN-BiLSTM narrows with horizon: 0.129 at $H{=}1$, 0.115 at $H{=}3$, and 0.097 at $H{=}5$. The per-lag median $\mathrm{F1}_+$ trajectories that underlie these aggregates are available in the artifact repository.

The original MAWIFlow benchmark [Schraven et al. 2026] reports markedly higher positive-class nAUT (e.g. XGBoost on all previous years, $\mathrm{nAUT}_1{=}0.752$; CNN-BiLSTM, 0.732). The two settings are *not directly comparable*: [Schraven et al. 2026] evaluates on a class-balanced, taxonomy/month-sampled subset with per-window HPO and single point estimates, whereas we evaluate on the native, highly imbalanced class distribution of our sampled annual snapshots with HPO-once and $R{=}5$ seed means, so positive-class F1, and hence nAUT, is measured under far lower anomaly prevalence and is structurally lower. The nAUT *definition* is identical (Eq. 1); the gap reflects the evaluation setting, not the metric. We therefore use [Schraven et al. 2026] as the dataset and protocol origin, not as a numerical baseline.

Sensitivity to the training-window regime. The fixed-window variants ($k{\in}\{1, 2, 3\}$) were also executed; these comparisons are descriptive, as no inferential tests were prespecified for them (mean nAUT_1 per regime and the per-seed artifacts are in the repository). The authors of MAWIFlow [Schraven et al. 2026] report two-to-three-year rolling windows as the strongest strategy in their setting; we retain the cumulative variant as primary because it matches the operational retrain-on-all-history scenario.

5. Discussion

Near-ceiling static performance is insufficient evidence of temporal robustness (RQ_1). On CICIoT2023, near-ceiling attack-class F1 gives the impression that model choice is inconsequential; in the complementary MAWIFlow evaluation the model families behave substantially differently under forward-chaining, with nAUT_1 ranging from 0.28 to 0.57—roughly a twofold spread. Near-ceiling performance on the static benchmark is therefore insufficient evidence of temporal robustness. Because the two axes differ in both dataset and domain, this contrast does not isolate the causal effect of the evaluation protocol; it is a deployment-oriented contrast rather than a controlled same-dataset ablation.

Sequential models obtain higher nAUT_H than tabular baselines in the cumulative MAWIFlow setting (H_1, consistently observed across $R{=}5$ seeds). The results em-

pirically support H_1 in the evaluated cumulative-window setting: both models operating on $W{=}16$ sequential windows achieved higher nAUT than the $1{\times}F$ tabular baselines across all five seeds (Table 6); the bootstrap 95% CIs are consistent with this ordering, the ranking is globally supported by a Friedman test ($p{=}0.002$), and one-sided pairwise Wilcoxon tests provide uncorrected directional evidence ($p{=}0.031$ per pair, $n{=}5$; none survives Bonferroni correction, $\alpha_c{=}0.0083$). The experiment does not, however, isolate the causal contribution of sequential context from architecture, capacity, optimization, or training dynamics; the empirical support for H_1 remains causally inconclusive, for the internal-validity reasons detailed in the Limitations below.

LightTransformer achieves the highest nAUT among the evaluated models (RQ_2). The LightTransformer's $R{=}5$ nAUT_1 advantage over CNN-BiLSTM (0.13 points at $H{=}1$, narrowing to 0.10 at $H{=}5$) is substantially larger than its F1_{macro} gap on CIC-IoT2023 (< 0.01 points), providing directional evidence for RQ_2: under the cumulative forward-chaining protocol, the LightTransformer achieved the highest nAUT among the evaluated model families and reported horizons, globally supported by Friedman and consistent across all five seeds. Notably, this lead is obtained under a markedly smaller HPO budget than the strongest tabular baseline (5 vs. 50 Optuna trials on Axis 2; Table 3), so the budget asymmetry runs against the LightTransformer rather than in its favor. Which architectural factor drives this lead—self-attention, causal masking, capacity, or training dynamics—cannot be determined from the current experiments (see Limitations); we therefore make no causal attribution.

XGBoost is strong on the static axis but less robust than sequential models under the cumulative MAWIFlow protocol. XGBoost achieves the best static performance ($\text{F1}_{\text{macro}}{=}0.928$) but ranks third on nAUT_1, consistent in this setting with reported limitations of gradient-boosted trees under distribution shift; receiving individual flow vectors, it cannot exploit within-window sequential context.

Limitations. *Construct validity:* the Axis-1 HPO objective (F1_+, chosen for alignment with nAUT_H) may under-calibrate the benign class, so the macro-F1 gap partially reflects this choice; per-class AUC-PR is left to future work. All models use a fixed 0.5 decision threshold, imbalance handling differs by family (inverse-frequency weights vs. tuned `scale_pos_weight`)—a non-controlled Axis-1 factor—and *lightweight* refers to parameter count and compactness only (efficiency metrics are not tabulated). *Internal validity:* the sequential-vs-tabular comparison varies input representation, architecture, capacity, and training dynamics simultaneously; the isolating ablation (LT at $W{=}1$) was not executed, so sequential context cannot be credited causally. Comparative conclusions are scoped to the cumulative window. *Statistical scope:* with five seeds, the smallest attainable one-sided Wilcoxon p ($0.03125{=}1/2^5$) is reached by all six pairs on nAUT_1, yet none survives Bonferroni correction ($\alpha_c{=}0.0083$), so the ranking has global Friedman support but no corrected pairwise confirmation; bootstrap CIs are indicative rather than tight; Friedman, CIs, and Wilcoxon derive from the same five runs (complementary, not independent); and seeds-as-blocks on a single dataset indicates cross-seed rank stability, not multi-dataset superiority. Results are statistically reproducible across seeds rather

than bit-for-bit (mixed-precision training, non-deterministic CUDA kernels).

6. Conclusion

We presented a comparative evaluation of four NIDS model families under static and temporal protocols.

RQ_1 *(addressed):* near-ceiling performance on the static CICIoT2023 benchmark is insufficient evidence of robustness under temporal drift: near-ceiling attack-class F1 coexists with a twofold spread in nAUT_1 (0.28–0.57) on MAWIFlow. As the axes differ in dataset and domain, this contrast is deployment-oriented evidence and does not isolate the protocol effect.

RQ_2 *and* H_1 *(cumulative training window; consistently observed across all four models,* $R{=}5$ *seeds):* LightTransformer ($\text{nAUT}_1{=}0.57$) leads CNN-BiLSTM (0.44), which leads XGBoost (0.38) and MLP (0.28); see Table 6 for confidence intervals. Bootstrap 95% CIs are consistent with this ordering, a Friedman test rejects equal rankings globally ($p{=}0.002$), and one-sided pairwise Wilcoxon tests provide uncorrected directional evidence ($p{=}0.031$ per pair, $n{=}5$); no pairwise comparison survives Bonferroni correction for six comparisons ($\alpha_c{=}0.0083$). These results empirically support H_1 under the reported cumulative-window protocol, with the causal Transformer achieving the highest nAUT among the evaluated model families (RQ_2). The conclusions are scoped to the cumulative strategy. The advantage is associated with sequential windows but not causally attributed to them; isolation requires the ablation noted below.

Our central methodological finding is that forward-chaining temporal evaluation should be a standard component of deployment-oriented NIDS benchmarking: it reveals robustness differences invisible to the static axis.

Future work: attention-map analysis of the temporal patterns driving robustness; cross-dataset generalization between MAWIFlow and CICIoT2023; a window-size ablation (LT with $W{\in}\{1, 4, 8, 16, 32\}$) to isolate sequential context from architectural and capacity effects; and inferential testing of the fixed-window regimes.

Use of Generative Artificial Intelligence

The authors used Claude Code (Sonnet 5 and Opus 5) and Codex (GPT 5.5) as generative AI tools to assist with textual review, structuring of ideas, and code development support. All AI-assisted content was critically reviewed and validated by the authors, who assume full responsibility for the accuracy, integrity, and originality of the final manuscript.

References

Abdulganiyu, O. H., Tchakoucht, T. A., and Saheed, Y. K. (2023). A systematic literature review for network intrusion detection system (ids). *International Journal of Information Security*, 22:1125–1162.

Akiba, T., Sano, S., Yanase, T., Ohta, T., and Koyama, M. (2019). Optuna. In *Proceedings of the 25th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 2623–2631. ACM.

Apruzzese, G., Pajola, L., and Conti, M. (2022). The cross-evaluation of machine learning-based network intrusion detection systems. *IEEE Transactions on Network and Service Management*, 19:5152–5169.

Chen, T. and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, volume 13-17-August-2016, pages 785–794. Association for Computing Machinery.

Chinnasamy, R., Subramanian, M., Easwaramoorthy, S. V., and Cho, J. (2025). Deep learning-driven methods for network-based intrusion detection systems: A systematic review. *ICT Express*, 11:181–215.

Fontugne, R., Borgnat, P., Abry, P., and Fukuda, K. (2010). Mawilab. In *Proceedings of the 6th International COnference*, pages 1–12. ACM.

Goldschmidt, P. and Chudá, D. (2025). Network intrusion datasets: A survey, limitations, and recommendations. *Computers & Security*, 156:104510.

Kheddar, H. (2025). Transformers and large language models for efficient intrusion detection systems: A comprehensive survey. *Information Fusion*, 124:103347.

Lansky, J., Ali, S., Mohammadi, M., Majeed, M. K., Karim, S. H., Rashidi, S., Hosseinzadeh, M., and Rahmani, A. M. (2021). Deep learning-based intrusion detection systems: A systematic review.

Lu, J., Liu, A., Dong, F., Gu, F., Gama, J., and Zhang, G. (2018). Learning under concept drift: A review. *IEEE Transactions on Knowledge and Data Engineering*, 31:1–1.

Manocchio, L. D., Layeghy, S., Lo, W. W., Kulatilleke, G. K., Sarhan, M., and Portmann, M. (2024). Flowtransformer: A transformer framework for flow-based network intrusion detection systems. *Expert Systems with Applications*, 241:122564.

Neto, E. C. P., Dadkhah, S., Ferreira, R., Zohourian, A., Lu, R., and Ghorbani, A. A. (2023). Ciciot2023: A real-time dataset and benchmark for large-scale attacks in iot environment. *Sensors*, 23.

Schraven, J., Windmann, A., and Niggemann, O. (2026). Mawiflow benchmark: Realistic flow-based evaluation for network intrusion detection. In *Proceedings of the 12th International Conference on Information Systems Security and Privacy*, pages 549–556. SCITEPRESS - Science and Technology Publications.

Sommer, R. and Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy*, pages 305–316. IEEE.

Wang, W., Zhu, M., Wang, J., Zeng, X., and Yang, Z. (2017). End-to-end encrypted traffic classification with one-dimensional convolution neural networks. In *2017 IEEE International Conference on Intelligence and Security Informatics (ISI)*, pages 43–48. IEEE.

Wu, Z., Zhang, H., Wang, P., and Sun, Z. (2022). Rtids: A robust transformer-based approach for intrusion detection system. *IEEE Access*, 10:64375–64387.

Yin, C., Zhu, Y., Fei, J., and He, X. (2017). A deep learning approach for intrusion detection using recurrent neural networks. *IEEE Access*, 5:21954–21961.

Zhou, H., Chen, J., Mei, Y., Adam, G., Aggarwal, V., Bastian, N. D., and Lan, T. (2024). Real-time network intrusion detection via importance sampled decision transformers. In *Proceedings - 2024 IEEE 21st International Conference on Mobile Ad-Hoc and Smart Systems, MASS 2024*, pages 82–91. Institute of Electrical and Electronics Engineers Inc.

Çağdaş Özer and Orman, Z. (2024). *Transformers Architecture Oriented Intrusion Detection Systems: A Systematic Review*, volume 1138 LNNS, pages 151–160. Springer Science and Business Media Deutschland GmbH.