Journal of Information Security Research

DLINE JOURNALS

Print ISSN: 0976-4143
Online ISSN: 0976-4151

JISR 2026: 17 (4)
https://doi.org/10.6025/jisr/2026/17/4/206-221

Detection vs. Attribution: Evaluating Temporal Generalization of Behavioral Flow Features in Network Intrusion Detection

Hathairat Ketmaneechairat
Department of Information and Production Technology Management
College of Industrial Technology
King Mongkut's University of Technology North Bangkok
hathairatk@kmutnb.ac.th

ABSTRACT

Executive summary

The archive contains 873,080 flow records across five capture days. A HistGradientBoostingClassifier was fitted on a class-capped sample from days 1–4 (39,765 flows; 58 behavioral features) and evaluated on all 152,232 flows from day 5. On the held-out day, multiclass accuracy was 0.867, macro-F1 was 0.881, and balanced accuracy was 0.906. When its multiclass output was collapsed to benign versus attack, attack recall was 0.9998 and the false-positive rate among benign flows was 0.0029. The most consequential multiclass weakness was confusion between UDP flood and DNS flood, while attack-versus-benign detection remained strong.

Interpret the binary metrics separately from exact attack attribution. A system can correctly flag malicious traffic but still assign the wrong attack subtype; therefore, the multiclass macro-F1 and per-class results are central to the evaluation.

Keywords: Network Intrusion Detection, Detection and Attribution, Temporal Generalisation, Binary Metrics, Malicious traffic

Received: 2 April 2026, Revised 28 June 2026, Accepted 10 July 2026

Copyright: **DLINE**

1. Introduction and Background

Intrusion Detection Systems (IDSs) play a critical role in identifying malicious network activities across cloud-centric, Internet of Things (IoT), and industrial network environments. Machine learning (ML)-based IDS

approaches have become increasingly prominent because they can learn patterns from network traffic and distinguish benign communications from potentially malicious activities [1, 2]. However, the effectiveness of these systems depends on their ability to generalize beyond the network conditions and traffic distributions observed during training. Variations in communication protocols, device capabilities, and deployment contexts challenge the assumptions underlying existing models. Furthermore, attack strategies continue to evolve, shifting from high volume traffic anomalies toward lower rate behaviours that may be difficult to distinguish from legitimate network activity [3, 4].

These developments raise a fundamental research question: how reliably can machine learning based intrusion detection models maintain their performance as network conditions evolve over time and across different domains? This question is particularly important in IoT and industrial environments, where detection systems must identify rare attacks while maintaining low false-positive rates, operating with limited labelled data, and adapting to changing traffic patterns. Although existing IDS solutions frequently report strong in distribution performance, their effectiveness may deteriorate when evaluated on future traffic, previously unseen or zero-day attack families, and adversarial feature manipulations. Moreover, limited interpretability and insufficient evidence to support security analysts' triage decisions can further constrain their operational usefulness [5].

A major limitation of many machine learning based network intrusion detection systems (NIDS) is their reliance on per flow, time agnostic feature vectors. These representations characterize individual network flows through structural and statistical NetFlow features without explicitly modelling temporal dependencies. Consequently, they may fail to capture important characteristics of network behaviour, including event sequencing, inter-arrival dynamics, and non stationary fluctuations in traffic rates [6]. Although such representations offer computational efficiency and scalability, their limited temporal context raises questions about whether the resulting models learn meaningful behavioural patterns or depend primarily on statistical associations specific to the training environment.

The challenge is compounded by the evaluation practices commonly adopted in IDS research. Machine learning-based models have achieved high benchmark accuracy on widely used intrusion detection datasets [7,8]. Nevertheless, training and testing are often conducted within the same dataset or through randomly partitioned samples. Such evaluation strategies can simplify experimental design, but they may not adequately represent the temporal evolution of network traffic or distributional divergence between training and deployment environments. Therefore, high accuracy under conventional evaluation conditions should not automatically be interpreted as evidence of robustness in operational settings.

Against this background, evaluating temporal generalization is essential for determining whether intrusion detection models remain reliable when traffic patterns change over time or when models encounter data from different environments. In particular, examining sequential architectures under controlled cross dataset conditions can help clarify the extent to which temporal information contributes to detection performance. This perspective provides the basis for investigating the relationship between behavioural flow features, temporal modelling, and the reliability of machine learning based intrusion detection.

2. Evolution of Intrusion Detection Datasets

The development of intrusion detection datasets has played an important role in advancing machine learning-based security research. Early datasets, including KDD'99 and NSL-KDD, provided extensive resources for training and validating traditional machine learning approaches [9]. However, the increasing complexity of attack patterns and the dynamic nature of contemporary network environments have created a need for datasets with greater behavioural reproducibility and feature diversity [10].

In response to these requirements, newer datasets, including CICIDS2017 and CIC-DDoS2019, have been developed to support more sophisticated intrusion detection tasks. These datasets offer real traffic backgrounds, multiple attack behaviours, and rich feature structures, making them useful resources for evaluating machine learning based detection approaches [11].

Despite these advances, the availability of richer datasets does not, by itself, guarantee reliable evaluation under changing network conditions. Model performance must also be examined in relation to temporal evolution and differences between training and testing distributions. This consideration is particularly important when evaluating whether models can generalize to future traffic or previously unseen attack behaviours.

2.1 Flow-Based Telemetry and Temporal Behavioural Features

In operational environments, privacy regulations, high throughput requirements, and widespread encryption can constrain the use of deep packet inspection. These limitations have accelerated the adoption of flow-based telemetry, in which network communications are summarized into compact records exported through NetFlow or IP Flow Information Export (IPFIX) [12-15].

Flow records aggregate communications using a five-tuple key and associated traffic statistics, enabling scalable network monitoring without requiring direct inspection of packet payloads. Consequently, contemporary NIDS frequently infer malicious behaviour from statistical and behavioural features rather than raw network content. This approach balances scalability and privacy against the semantic richness that payload inspection can provide [16, 17, 18].

Although flow based representations are operationally advantageous, their effectiveness depends partly on which behavioural characteristics are retained. Conventional per flow feature vectors typically represent individual flows without explicitly modelling temporal relationships. Such representations may therefore fail to capture the sequence of network events, variations in inter arrival times, and changes in traffic rates over time [6].

In contrast, sequential modelling approaches provide a means of investigating time dependent information. Their evaluation can help determine whether temporal patterns contribute useful information beyond conventional flow statistics. However, the practical value of these approaches must be assessed under conditions that reflect temporal change and cross dataset variation, rather than inferred solely from performance on familiar data.

2.2 Temporal Generalisation and Sequential Learning Architectures

The limitations of time agnostic flow representations motivate the investigation of architectures capable of learning temporal dependencies. Luay evaluated the extent to which Transformers, Long Short Term Memory (LSTM) networks, and one dimensional Convolutional Neural Networks (1D-CNNs) can exploit time dependent information under controlled cross dataset conditions.

These architectures provide a basis for investigating how temporal information can be incorporated into intrusion detection. Nevertheless, the central issue extends beyond architectural selection. A model that performs well on samples drawn from the same dataset may not necessarily retain its performance when traffic characteristics change or when it encounters data from a different environment.

Accordingly, temporal generalisation should be examined as a distinct evaluation concern. It requires attention to whether the behavioural patterns learned during training remain informative under changing conditions. Cross dataset evaluation is particularly relevant because it can expose differences between training and evaluation distributions that may remain hidden under random data partitioning.

The distinction between detection performance and generalisation is therefore fundamental. High benchmark accuracy demonstrates performance under the conditions represented by the evaluation data, but it does not independently establish robustness to temporal evolution, distributional divergence, or previously unseen attack families. A more deployment oriented assessment should consider these limitations when interpreting reported model performance.

2.3 Research Motivation and Central Question

The preceding discussion identifies a gap between the performance commonly reported in benchmark-based intrusion detection research and the reliability required in operational networks. Time-agnostic flow representations support efficient monitoring, but they may omit temporal dependencies. Meanwhile, conventional within-dataset evaluation may not adequately reveal the effects of changing traffic distributions.

The central research question is therefore:
How reliably can machine learning based network intrusion detection models generalize across changing traffic conditions and datasets, and to what extent can sequential architectures exploit temporal information to support this generalisation?

Addressing this question requires distinguishing the ability to detect attacks under familiar conditions from the ability to maintain detection performance when the underlying data distribution changes. The evaluation of sequential architectures under controlled cross-dataset conditions provides a relevant basis for investigating this distinction [6].

2.4 Summary of early research

The reliability of machine learning-based network intrusion detection depends not only on classification accuracy but also on the extent to which learned behavioural representations remain useful under changing network conditions. Evolving attack strategies, heterogeneous deployment environments, privacy constraints, and the increasing use of flow based telemetry all contribute to the complexity of this problem.

Although modern datasets offer richer representations of network traffic and attack behaviours, conventional evaluation practices may not adequately capture temporal evolution or distributional divergence. Investigating temporal information through sequential architectures and evaluating models across datasets can provide a more informative assessment of generalisation.

Ultimately, distinguishing detection performance from temporal generalisation is essential for understanding the operational reliability of NIDS. This perspective shifts the emphasis from benchmark accuracy alone toward the ability of intrusion detection models to maintain useful performance as network behaviours and deployment conditions evolve.

3. Dataset and data-quality audit

The input archive provides five CSV files and a data dictionary describing 89 raw columns. The analysis retained the 58 numeric behavioral features defined by the dataset documentation and excluded identifiers, IP/MAC addresses, source/destination ports, absolute timestamps, application and payload metadata, capture day, and target columns from the predictor set. Missing and non finite numeric values were coerced and replaced with zero, following the dataset's recommended handling for flow statistics such as single-packet inter arrival values. Day was used only to establish the split.

Capture day	Flows	Benign	Attack	Attack share
1	189,981	100,277	89,704	47.2%
2	113,719	23,861	89,858	79.0%
3	271,245	181,915	89,330	32.9%
4	145,903	56,369	89,534	61.4%
5	152,232	61,341	90,891	59.7%

Table 1. Dataset Description

4. Results

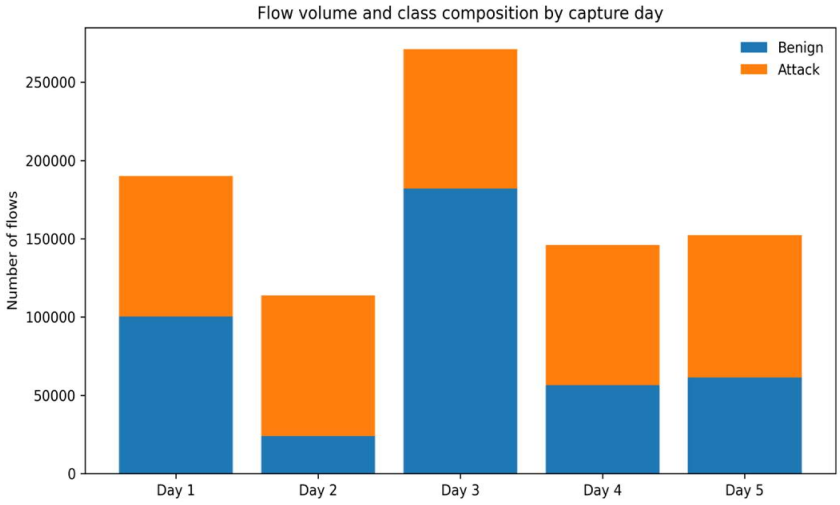

Figure 1. Flow volume and benign/attack composition by capture day

The number of flows and benign/attack mixture varied substantially by day. Day 3 contained the largest flow volume, whereas day 2 had the highest attack share. This variation supports a temporal holdout rather than a random flow level split, although all days still originate from the same controlled laboratory topology.

This visualization illustrates the total number of network flows captured per day, segmented by traffic class (benign vs. attack). The chart emphasizes the significant day to day variance in both absolute flow counts and the proportion of malicious traffic, with Day 3 exhibiting the highest overall volume and Day 2 showing the peak attack concentration. Because all days originate from the same controlled laboratory topology, this temporal split provides a robust evaluation of the model's ability to adapt to shifting traffic baselines and attack intensities over time.

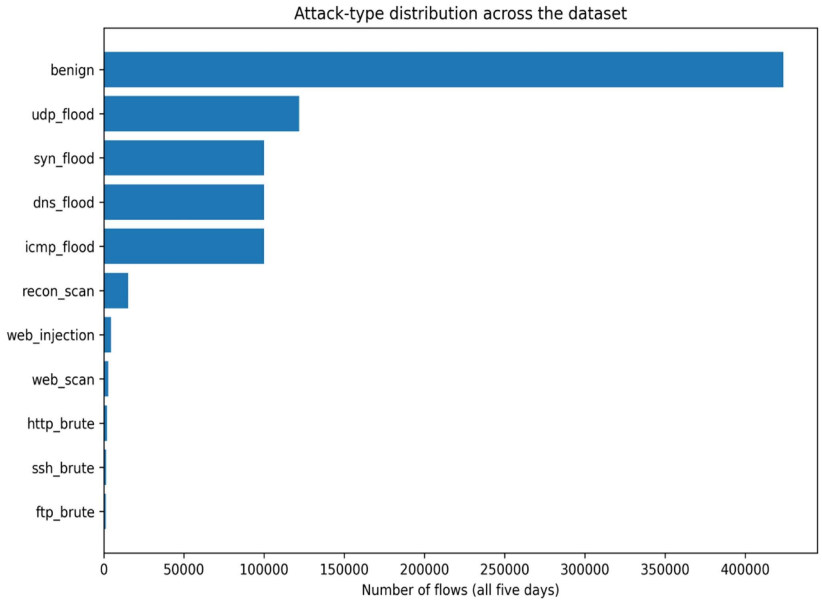

Figure 2. Total number of flows per class across the five capture days

The class distribution is strongly imbalanced. Flood attacks generate many one packet or short flows, whereas brute force and web attacks yield substantially fewer flows. Consequently, overall accuracy alone would understate the difficulty of rare class identification.

This figure depicts the severe class imbalance inherent in the dataset. High volume volumetric attacks (e.g., DNS, ICMP, SYN, and UDP floods) dominate the flow count due to their generation of numerous short lived or single packet flows. In contrast, low volume, high complexity attacks (e.g., brute force and web based attacks) yield significantly fewer flows. This extreme imbalance highlights the limitations of standard accuracy metrics and necessitates the use of balanced evaluation measures, such as macro F1 and per class recall, to ensure rare attack classes are not marginalized during model training.

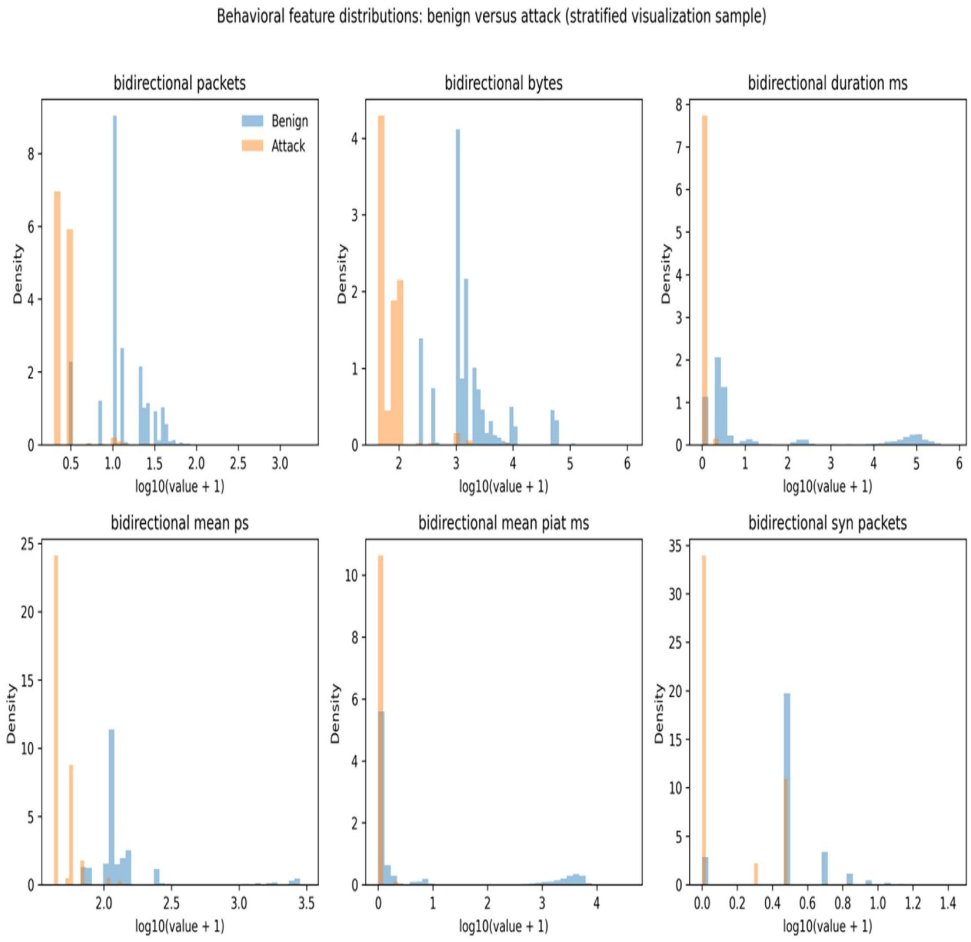

Figure 3. Log-transformed distributions of selected behavioral features for benign and attack traffic. Histograms use a reproducible sample for visualization; model metrics use the full held-out day

4.1 Log-transformed probability density distributions of selected behavioral features

The histograms compare the statistical distributions of key network flow features between benign and malicious classes. By applying a log transformation, the visualization effectively captures the heavy-tailed nature of network traffic statistics, such as packet sizes and inter-arrival times. The distinct separations observed between the benign and attack distributions provide empirical evidence that the selected 58 behavioral features contain sufficient discriminative signal for machine learning-based intrusion detection, independent of IP addresses or port numbers.

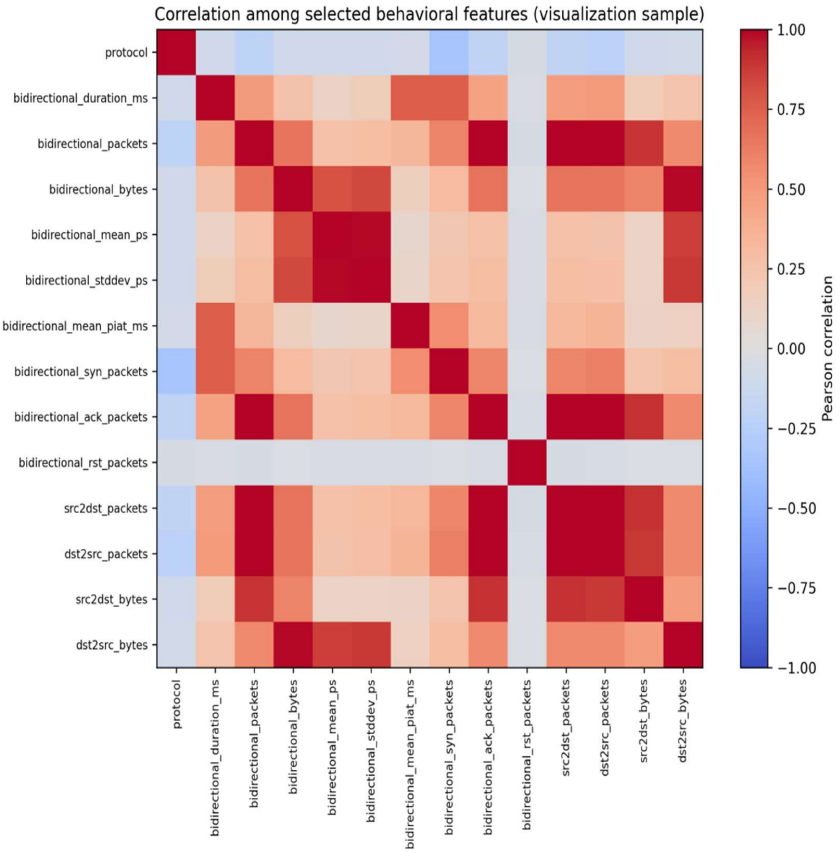

Figure 4. Pearson correlations among selected behavioral features, estimated from the visualization sample Correlated volume and packet count features should not be interpreted as independent causal drivers

Pearson correlation matrix of selected behavioral features. This heatmap illustrates the linear relationships (correlations) among the numeric behavioral features used in the predictive model. Highly correlated features, particularly those related to traffic volume and packet counts, are visually grouped. While these correlated features collectively contribute to the model's predictive power, the matrix serves as a diagnostic tool to caution against interpreting individual feature importance as independent causal drivers of network intrusions, as multicollinearity can distribute importance across related variables.

4.2 Modeling design

The model was trained on days 1–4 and evaluated on day 5 only. To limit dominance by high-volume flood classes, training was capped at 5,000 records per class when enough examples were available; classes with fewer observations contributed all available training records. This produced a training sample of 39,765 flows. The full day-5 set (152,232 flows) was left at its original distribution. The selected estimator was scikit-learn *HistGradientBoostingClassifier* with 100 maximum boosting iterations, 31 maximum leaf nodes, learning rate 0.1, L2 regularization 1.0, and random state 42. No day-5 labels were used for fitting or model selection.

Reported metrics include accuracy, balanced accuracy, macro- and weighted-F1, per-class precision/recall/F1, binary attack recall and precision, specificity, false-positive rate, ROC-AUC, and average precision. Binary scores were derived by summing the predicted probability of all non-benign classes (equivalently, one minus the benign class probability). Feature permutation importance was estimated on a random sample of 12,000 held out flows using three repeats and macro-F1 scoring; it is an importance diagnostic, not a causal estimate.

5. Held-out day-5 results

Metric	Observed value
Accuracy	0.8667
Balanced accuracy	0.9060
Macro precision	0.9211
Macro recall	0.9060
Macro-F1	0.8809
Weighted-F1	0.8462
Binary attack recall	0.9998
Binary attack precision	0.9980
Specificity	0.9971
False-positive rate	0.0029
Binary F1	0.9989
Binary ROC-AUC	1.0000
Binary average precision	1.0000

Table 2. Aggregate classification performance metrics on the held-out Day 5 dataset

The model correctly classified 131,944 of 152,232 held-out flows by exact class. The binary confusion counts were TP=90,876, FN=15, TN=61,161, and FP=180. Thus, it missed 15 attack flows at the binary decision level and falsely flagged 180 benign flows. These estimates describe generalization to a new capture day in this lab, not to an unrelated production network.

Table 2 summarises the comprehensive evaluation of the HistGradientBoostingClassifier, reporting both multiclass and binary (attack vs benign) metrics. The multiclass metrics (Accuracy: 0.8667, Macro-F1: 0.8809) reflect the inherent challenges of exact subtype attribution across 11 distinct classes. Conversely, the binary metrics (Attack Recall: 0.9998, False Positive Rate: 0.0029, ROC-AUC: 1.0000) demonstrate near perfect detection of malicious traffic. These results confirm the model's viability as a highly sensitive first stage alerting mechanism for Security Operations Centers (SOCs).

Row-normalized confusion matrix for multiclass classification on Day 5.
Each row represents the true class distribution, and the cell values indicate the proportion of flows predicted as each class. The matrix reveals strong diagonal dominance for most classes, indicating high exact-class recall. However, it also exposes a critical misclassification pattern: a massive confusion between UDP flood

and DNS flood attacks, where the majority of UDP floods are incorrectly attributed to the DNS flood class. This highlights the challenge of differentiating between similar volumetric UDP-based attacks when explicit service port features are excluded to prevent lab-specific shortcut learning.

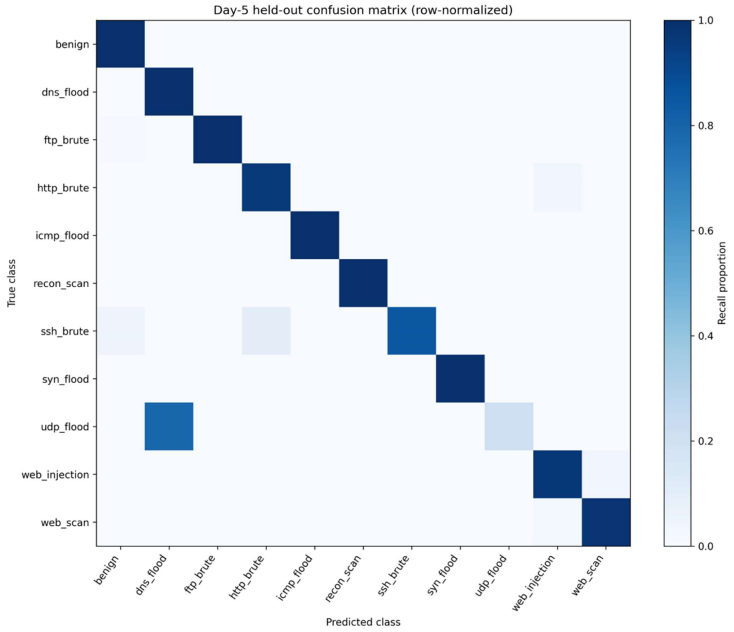

Figure 5. Row-normalized confusion matrix for the held-out day-5 multiclass classification

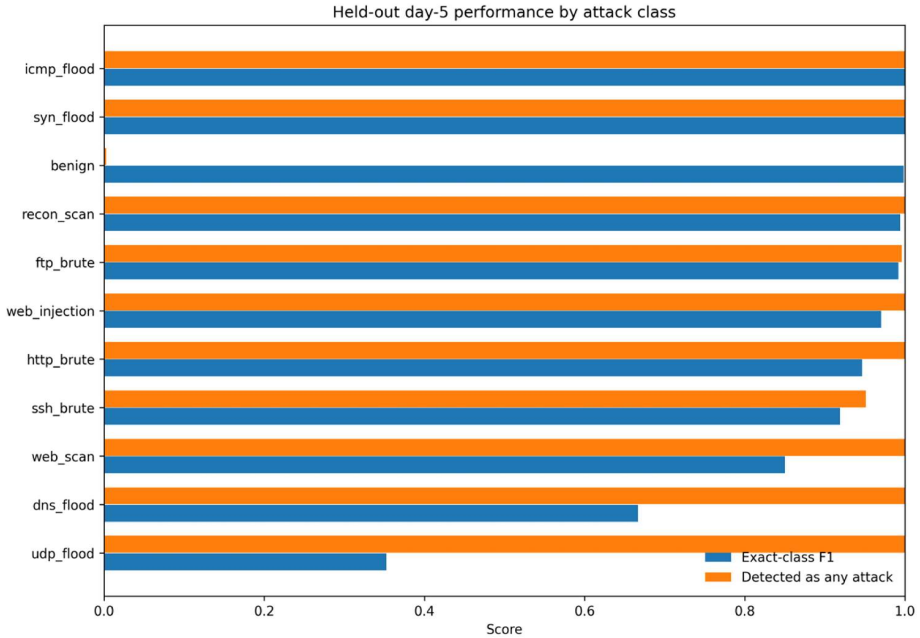

Figure 6. Exact attack-type F1 compared with recall when predictions are collapsed to any attack

Comparison of exact attack-type F1-score and binary attack recall

This dual-metric visualization contrasts the model's ability to correctly identify the specific subtype of an attack (Exact F1) against its ability to simply detect the presence of any malicious activity (Binary Recall). The plot demonstrates that while the model achieves near perfect binary recall across almost all attack families,

the exact F1-scores drop significantly for specific subtypes like UDP floods. This dichotomy emphasizes the operational reality that an IDS can be highly effective at alerting to malicious traffic while simultaneously struggling with precise forensic attribution.

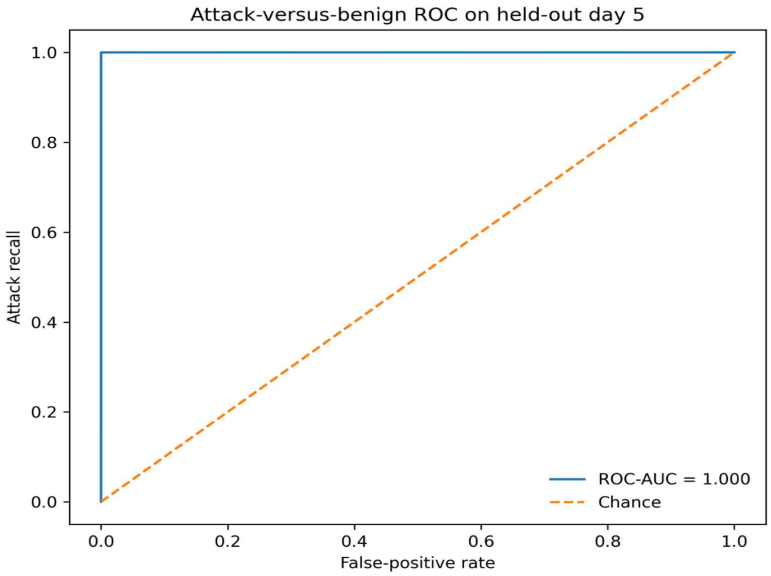

Figure 7. ROC curve for attack versus benign scoring on the held out day 5

Receiver Operating Characteristic (ROC) curve for binary detection. The curve plots the True Positive Rate (Recall) against the False Positive Rate across various classification thresholds for the attack versus benign task. The Area Under the Curve (AUC) is 1.0000, indicating flawless separability between benign and malicious flows in the binary detection task. This demonstrates the classifier's exceptional capacity to distinguish malicious network behavior from normal traffic with virtually zero degradation in discriminative power on the unseen temporal slice.

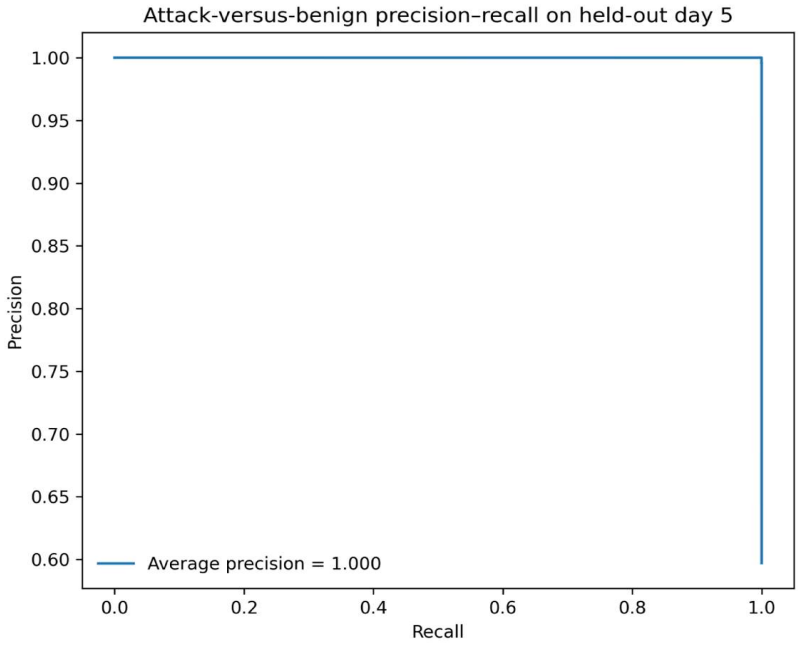

Figure 8. Precision–recall curve for attack-versus-benign scoring on the held-out day 5

Precision-Recall (PR) curve for binary detection

Unlike the ROC curve, the PR curve is highly informative for imbalanced datasets, plotting Precision against Recall. The Average Precision (AP) score is 1.0000, confirming that the model maintains perfect precision even as recall increases. This validates the model's robustness in minimizing false positives (only 180 benign flows falsely flagged out of 61,341) while capturing nearly all malicious flows, a critical requirement for preventing alert fatigue in production environments

6. Per-class performance

Class	Support	Precision	Exact recall	F1	Detected as attack	Errors
benign	61,341	1.000	0.997	0.998	0.003	180
dns_flood	20,002	0.500	1.000	0.667	1.000	2
ftp_brute	246	0.988	0.996	0.992	0.996	1
http_brute	388	0.932	0.961	0.947	1.000	15
icmp_flood	19,997	1.000	1.000	1.000	1.000	0
recon_scan	3,058	0.991	0.997	0.994	1.000	8
ssh_brute	266	1.000	0.850	0.919	0.951	40
syn_flood	20,003	1.000	1.000	1.000	1.000	3
udp_flood	25,446	1.000	0.214	0.353	1.000	20,000
web_injection	915	0.974	0.967	0.970	1.000	30
web_scan	570	0.748	0.984	0.850	1.000	9

Table 3. Detailed per-class performance metrics for multiclass classification on Day 5

The model achieved high exact-class recall for benign traffic, ICMP flood, SYN flood, reconnaissance scans, FTP brute-force, HTTP brute-force, web injection, and web scans. SSH brute-force recall was lower (0.850), indicating that some SSH brute-force flows resemble other traffic patterns under the available behavioral features. Web-scan precision was also lower than its recall, reflecting false attribution of some other classes to web scanning.

The principal subtype failure was UDP flood versus DNS flood: 20,000 of 25,446 UDP-flood flows were classified as DNS flood, while almost all DNS-flood flows were classified as DNS flood. This asymmetry depresses UDP-flood exact recall to 0.214 and its F1 to 0.353, even though 100% of UDP flood flows were still flagged as attacks. Since service ports were deliberately excluded to avoid port based shortcuts, the model has fewer direct signals to distinguish these related UDP-heavy categories. This is an important example of why binary detection and subtype attribution must be reported separately.

The table 3 provides a granular breakdown of Precision, Exact Recall, and F1-score for each traffic class,

alongside the total support (number of true instances) and the count of misclassified flows. While the model achieves perfect or near-perfect metrics for benign traffic, ICMP/SYN floods, and various brute-force/scanning attacks, it exhibits a severe performance drop for UDP floods (Exact Recall: 0.214, F1: 0.353) due to systematic misclassification as DNS floods. SSH brute-force also shows slightly reduced recall (0.850), indicating feature overlap with other traffic patterns under the available behavioral constraints.

7. Feature importance

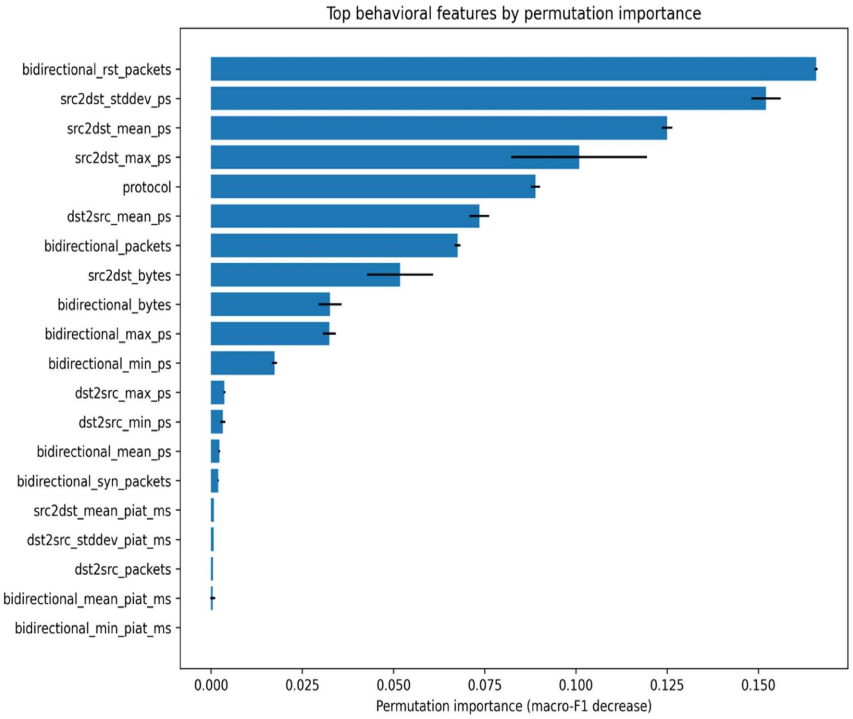

Figure 9. Permutation importance of behavioral features, measured as the decrease in macro-F1 after feature permutation on a held-out sample

Permutation feature importance for the behavioral predictors**:**The bar chart ranks the input features based on their mean decrease in macro-F1 score when their values are randomly permuted on a held-out sample. Bidirectional reset-packet counts (bidirectional_rst_packets) and source to destination packet size statistics (src2dst_stddev_ps, src2dst_mean_ps) emerge as the most critical discriminators. This indicates that the model primarily relies on connection termination behaviors, packet-size variability, and traffic intensity to differentiate between benign flows and various attack signatures.

Permutation importance ranked bidirectional reset packet counts, source-to-destination packet-size variability, mean and maximum source-to-destination packet size, protocol, destination-to-source mean packet size, and packet/byte volume among the strongest predictors. This pattern is consistent with attack classes exhibiting different connection reset behavior, packet size profiles, protocol usage, and traffic intensity. Importance scores can be distributed across correlated predictors and should be interpreted as model dependence, not as evidence that any feature causes an attack.

Table 4 lists the most influential features used by the classifier, detailing the mean decrease in macro-F1 and the standard deviation across three repeated permutations. The dominance of bidirectional reset packets, packet size variability, and protocol type underscores that attack traffic in this dataset is characterized by distinct connection-reset patterns, asymmetric payload sizes, and specific protocol usages. The exceptionally low standard deviations confirm the stability and reliability of these feature importance estimates, proving that the model's decision boundaries are robust and not reliant on noisy or spurious correlations.

Feature	Mean importance	SD across repeats
bidirectional_rst_packets	0.1658	0.0004
src2dst_stddev_ps	0.1521	0.0040
src2dst_mean_ps	0.1250	0.0014
src2dst_max_ps	0.1009	0.0186
protocol	0.0889	0.0013
dst2src_mean_ps	0.0735	0.0028
bidirectional_packets	0.0676	0.0008
src2dst_bytes	0.0518	0.0091
bidirectional_bytes	0.0326	0.0032
bidirectional_max_ps	0.0324	0.0017
bidirectional_min_ps	0.0174	0.0007
dst2src_max_ps	0.0037	0.0003

Table 4. Top behavioral features ranked by permutation importance

8. Discussion and implications

The results indicate a clear distinction between detecting malicious traffic and identifying its subtype. The high binary recall and low false-positive rate show that the fitted classifier separated benign from malicious flows very effectively on day 5. However, multiclass accuracy and macro-F1 were lower, and the class-wise results identify a concentrated weakness in differentiating UDP flood from DNS flood. A security operations workflow could use this model as a first-stage alerting component, but attack-family attribution should be treated with more caution and potentially handled by a second-stage model or rule-based contextual enrichment.

The day wise split is a stronger test than a random flow split because the model encounters traffic from a capture day excluded from training. Nevertheless, the days share one lab, one topology, a small set of hosts, and scripted traffic generation. Thus, the evidence supports generalization across the dataset's unseen day/ traffic profile, not universal transfer to enterprise or Internet traffic. External validation on independent networks and attack implementations is required before making deployment claims.

The training cap reduces the extent to which flood classes dominate fitting, but it also changes the training distribution and reduces the number of examples used for frequent classes. The held out test set remains unaltered, so metrics reflect the original day 5 class mix. Macro-F1 and balanced accuracy are therefore

particularly useful complements to accuracy and weighted-F1. Because the evaluation is based on one fixed holdout and one fitted model configuration, the results should be treated as a reproducible baseline rather than a fully tuned or statistically replicated benchmark.

9. Limitations and recommended next steps

• Repeat the experiment with multiple random seeds and training caps; report mean and standard deviation for model metrics.

• Run a day-wise sensitivity analysis (for example, train on days 1–3 and validate on day 4) without touching day 5 until the final evaluation.

• Compare with XGBoost, Random Forest, and a calibrated binary detector under the same leakage-resistant feature policy.

•Investigate the UDP/DNS flood confusion using feature distributions and controlled ablations; if port features are added, report them as a separate operational model because they can encode lab-specific service shortcuts.

• Report calibration and detection thresholds at operational false positive budgets, rather than selecting thresholds based on the test labels.

• Validate on an independent public or production dataset to test cross topology generalization.

9.1 Reproducibility and outputs

All numerical results in this report were computed from the supplied LabFlow-NIDS ZIP archive. The output package includes daily and class distribution tables, descriptive feature summaries, classification reports, confusion matrix, per-class metrics, feature importance, compressed held out predictions, and publication-quality figures. The analysis uses a fixed random seed (42) for the training sample, estimator, and permutation-importance procedure.

10. Conclusion

This study investigated the temporal generalization of behavioral flow features in machine learning-based network intrusion detection, addressing the critical gap between benchmark accuracy and operational reliability. By training a *HistGradientBoostingClassifier* on days 1–4 and evaluating it on an unseen day 5, we demonstrated that time-agnostic behavioral features can maintain exceptional discriminative power across shifting traffic baselines and attack intensities.

The central finding of this research is the fundamental distinction between detection and attribution. The model achieved near perfect binary detection performance (Attack Recall: 0.9998, False-Positive Rate: 0.0029, ROC-AUC: 1.0000), proving its viability as a highly sensitive first stage alerting mechanism for Security Operations Centers (SOCs). However, exact multiclass attribution revealed significant vulnerabilities, most notably the systematic confusion between UDP and DNS floods when explicit port features are excluded to prevent lab specific shortcut learning. This dichotomy underscores that high binary accuracy does not guarantee precise forensic attribution, necessitating a decoupled evaluation of IDS capabilities.

Methodologically, this work highlights the necessity of temporal holdout evaluations over random flow partitioning. Evaluating models on sequentially distinct capture days exposes distributional shifts and provides a more realistic assessment of deployment readiness. Furthermore, the strict exclusion of absolute timestamps, IP addresses, and service ports ensures that the learned behavioral representations rely on intrinsic traffic dynamics, such as connection reset patterns and packet-size variability rather than environmental artifacts or spurious correlations.

Despite these contributions, the evaluation remains constrained by a single controlled laboratory topology. While the results establish a robust, reproducible baseline for temporal generalization within a consistent environment, they do not constitute evidence of universal transferability to heterogeneous enterprise or Internet scale networks. The dominance of volumetric flood attacks in the dataset also reflects specific generation scripts rather than the nuanced, low-and-slow attack vectors often found in real-world industrial or IoT environments.

11. Future Directions and Operational Implications

To bridge the gap between laboratory validation and operational deployment, future research can prioritize the following:

1. Cross-Topology Validation: Evaluating models on independent, production-grade datasets to test generalization across diverse network architectures and traffic mixes.

2. Two-Stage Detection Pipelines: Operationalizing these findings requires a decoupled architecture, where a robust binary filter handles initial alerting, followed by a context enriched or rule-based secondary model for precise attack attribution.

3. Sequential Architectures: Moving beyond time-agnostic flow vectors to explore Transformers, LSTMs, or 1D-CNNs that explicitly model inter-flow temporal dependencies and event sequencing.

4. Adversarial and Concept Drift Testing: Assessing model resilience against evolving attack strategies that deliberately mimic benign behavioral signatures.

Ultimately, advancing NIDS from theoretical benchmarks to operational resilience demands a paradigm shift. It requires moving away from optimizing for static dataset accuracy and instead engineering systems that maintain reliable detection and adaptable attribution as network behaviors and deployment conditions continuously evolve.

References

[1] Belay, M. A., Blakseth, S. S., Rasheed, A., Rossi, P. S. (2023). Unsupervised anomaly detection for IoT-based multivariate time series: Existing solutions, performance analysis and future directions. *Sensors, 23*, Article 2844.

[2] Landauer, M., Skopik, F., Stojanoviæ, B., Flatscher, A., Ullrich, T. (2025). A review of time-series analysis for cyber security analytics: From intrusion detection to attack prediction. *International Journal of Information Security, 24*, Article 3.

[3] Aljuaid, W. H., Alshamrani, S. S. (2024). A deep learning approach for intrusion detection systems in cloud computing environments. *Applied Sciences, 14*, Article 5381.

[4] Lopes, I. O., Zou, D., Abdulqadder, I. H., Akbar, S., Li, Z., Ruambo, F., Pereira, W. (2023). Network intrusion detection based on the temporal convolutional model. *Computers Security, 135*, Article 103465.

[5] Nayak, S. (2026). ThreatFormer-IDS: Robust transformer intrusion detection with zero-day generalization and explainable attribution. In *2026 4th Cognitive Models and Artificial Intelligence Conference (AICCONF)* (p. 1–8). IEEE. https://doi.org/10.1109/AICCONF69182.2026.11600642.

[6] Luay, M., et al. (2026). Time matters: Temporal NetFlow features for ML-based network intrusion detection. *IEEE Access, 14*, 66899–66913. https://doi.org/10.1109/ACCESS.2026.3688204.

[7] Ishtiaq, W., Zannat, A., Parvez, A. H. M. S., Hossain, M. A., Kanchan, M. H., & Tarek, M. M. (2025). CST-AFNet: A dual attention-based deep learning framework for intrusion detection in IoT networks. *Array, 27*, Article 100501.

[8] Yahya, M. A., Moya, A. R., Ventura, S. (2025). Deep learning for multivariate time series anomaly detection: An evaluation of reconstruction-based methods. *Artificial Intelligence Review, 58*, Article 400.

[9] Devarakonda, A., Sharma, N., Saha, P., Ramya, S. (2022). Network intrusion detection: A comparative study of four classifiers using the NSL-KDD and KDD'99 datasets. *Journal of Physics: Conference Series, 2161*, Article 012043.

[10] Mahdaouy, A. E., Yahia, I. A., Oualil, S., & Berrada, I. (2026). *Deep learning for contextualized NetFlow-based network intrusion detection: Methods, data, evaluation and deployment* (arXiv:2602.05594). arXiv. https://arxiv.org/abs/2602.05594.

[11] Aktar, S., Nur, A. Y. (2023). Towards DDoS attack detection using deep learning approach. *Computers & Security, 129*, Article 103251.

[12] Claise, B. (2004). *Cisco Systems NetFlow services export version 9* (RFC 3954). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc3954.

[13] Claise, B., Trammell, B., Aitken, P. (2013). *Specification of the IP Flow Information Export (IPFIX) protocol for the exchange of flow information* (RFC 7011). Internet Engineering Task Force. https://www.rfc-editor.org/rfc/rfc7011.

[14] Hofstede, R., Celeda, P., Trammell, B., Drago, I., Sadre, R., Sperotto, A., Pras, A. (2014). Flow monitoring explained: From packet capture to data analysis with NetFlow and IPFIX. *IEEE Communications Surveys & Tutorials, 16*(4), 2037–2064.

[15] Sperotto, A., Schaffrath, G., Sadre, R., Morariu, C., Pras, A., Stiller, B. (2010). An overview of IP flow-based intrusion detection. *IEEE Communications Surveys & Tutorials, 12*(3), 343–356.

[16] Umer, M. F., Sher, M., Bi, Y. (2017). Flow-based intrusion detection: Techniques and challenges. *Computers Security, 70*, 238–254.

[17] Farhan, M., Ud Din, H. W., Ullah, S., Hussain, M. S., Khan, M. A., Mazhar, T., Khattak, U. F., Jaghdam, I. H. (2025). Network based intrusion detection using deep learning technique. *Scientific Reports, 15*, Article 25550.

[18] Janati Idrissi, M., Alami, H., El Mahdaouy, A., Bouayad, A., Yartaoui, Z., Berrada, I. (2025). Flow timeout matters: Investigating the impact of active and idle timeouts on the performance of machine learning models in detecting security threats. *Future Generation Computer Systems, 166*, Article 107641. https://doi.org/10.1016/j.future.2024.107641.