Auditable Flood Attack Detection using Isolation Forest with Decision Predicate Graphs

Eron Ponce Pereira[1], **Bruno Bogaz Zarpelão**[1], **Sylvio Barbon Junior**[2]

[1]Computer Science Department, State University of Londrina, Londrina-PR, Brazil

[2]Department of Engineering and Architecture, University of Trieste, Trieste, Italy

{eron.ponce.pereira,brunozarpelao}@uel.br, sylvio.barbonjunior@units.it

Abstract. *In computer networks, anomaly alerts are frequent, but there is an explainability gap between an anomaly alert and the observable network evidence needed for triage and incident reporting. We evaluate Isolation Forest on CICIoT2023 IoT traffic for DoS/DDoS flood attacks and interpret the learned trees with Decision Predicate Graphs (DPGs). Using fixed benign baselines, we test a Single, Dual, and Triple Attack Scenario while sweeping the attack fraction rate. We report precision, recall, F1 and error rates on fixed test sets, and use DPG predicates and co-occurrence relations to trace how network flows become outliers, turning model decisions into auditable network-level conditions that remain interpretable under mixed attacks and contaminated training.*

1. Introduction

The increasing complexity of Machine Learning (ML) models, although yielding high detection rates against network attacks, often turns them into non-interpretable models (black boxes), making it difficult to understand how a model is behaving [Gaitan-Cardenas et al. 2023] [Nwakanma et al. 2023] [Johnstone and Akinfaderin 2025]. This lack of transparency limits trust and practical applicability for security operators and raises additional concerns in security-critical systems.

In modern IoT networks, anomaly-based intrusion detection is an attractive alternative to fully supervised Intrusion Detection Systems (IDS) because operators often face scarce or unreliable labels and rapidly evolving traffic patterns, making unsupervised methods valuable for flagging suspicious flows at scale [Mutambik et al. 2024]. Among these, Isolation Forest (IF) is widely adopted due to its efficiency and scalability in unlabeled settings: it detects anomalies by recursively partitioning the feature space with random splits, so outliers are typically isolated with fewer splits (shorter path lengths) than inliers [Liu et al. 2008]. However, because this isolation process depends on randomized feature and split-point selection, alerts can be difficult to justify in operational terms, motivating the need for explanations that make IF decisions traceable and analyst-actionable.

In practice, this creates an auditability gap beyond detection rates, making explanations necessary to support analyst triage, validate detected anomalies, and guide tuning by revealing potential biases or errors induced by this randomness. To address this need, Explainable Artificial Intelligence (XAI) has emerged as a research area dedicated

to explaining ML models, with the goal of making these models more transparent and understandable to humans [Nwakanma et al. 2023].

Authors such as [Zouhri et al. 2024] and [Afnan Birahim et al. 2025] discuss explainability strategies for intrusion detection, with many relying on post hoc methods that are widely adopted in practice. In this line, SHapley Additive exPlanations (SHAP) [Bacevicius et al. 2025, Gaitan-Cardenas et al. 2023] is commonly used to produce per-instance attributions that are often aggregated to summarize feature importance, and Local Interpretable Model-agnostic Explanations (LIME) is also frequently applied for local, instance-level explanations. However, because these approaches explain the output after the fact and depend on perturbations, sampling, or background choices, their explanations can be unstable and difficult to translate into auditable decision evidence.

This limitation is especially constraining in anomaly-based intrusion detection, where operators need clear, defensible reasons for why a flow was flagged. Decision Predicate Graphs (DPG) [Arrighi et al. 2024, Ceschin et al. 2025] address this need by providing model-faithful, audit-ready explanations that follow the detector's own decision logic and yield traceable evidence to justify alerts.

In this paper, we evaluate how DPG can make IF alerts auditable for IoT flood detection. Using CICIoT2023, we conduct a controlled benchmark with benign baselines and single-, dual-, and triple-attack mixtures under a sweep of training attack ratios, and we jointly analyze detection outcomes and DPG evidence to assess how training conditions affect both discrimination and the resulting decision rationale. Since DPG is extracted from the trained model and does not change its predictions, performance results are reported to contextualize the auditability analysis across settings. Our main contributions are:

- A controlled CICIoT2023 benchmark for IoT flood anomaly detection with benign baselines, single/dual/triple mixtures, and a sweep of attack fraction injected in training.
- An audit-focused interpretation of IF using DPG, exposing predicate constraints and their co-occurrence as a transparent layer over the trained detector.
- A structured DPG pipeline with predicate ranking based on propagation behavior, co-occurrence patterns, betweenness-based bridges, and community modules (backbone + label propagation) to summarize model logic.
- A comparison with SHAP and LIME, showing similar feature families while DPG adds interaction and path-level structure beyond attribution.

The research questions target three practical concerns in auditable flood attack as anomaly detection. RQ1: How do training contamination and scenario composition (clean vs. contaminated; single vs. dual vs. triple mixtures) affect IF detection performance and score-space separability, and under which regimes does malicious exposure improve discrimination versus normalizing attack patterns? RQ2: Which predicate structures does the DPG reveal on top of IF, and are the resulting predicate thresholds and constraints consistent with flood-style attack semantics and expert expectations? RQ3: Does adding DPG explanations improve interpretability without degrading detection performance when compared with widely adopted state-of-the-art explainers? The experimental design addresses each question directly, and the Results section is organized accordingly.

The paper is organized as follows: Section 2 reviews related work; Section 3 describes the dataset, scenarios, contamination protocol, IF setup, and DPG pipeline; Section 4 presents results on detection and interpretability; and Section 5 concludes with future work.

2. Related Work

Explainable Artificial Intelligence (XAI) has grown substantially in recent years, although concerns about transparency in computational systems are older [Héder 2023, Gunning et al. 2021]. In anomaly-based intrusion detection for IoT networks, explainability is not only a general need, but an operational requirement, since alerts should be traceable to observable network evidence that supports analyst triage and incident reporting. This is particularly relevant for IF, which remains attractive under scarce or unreliable labels, but can yield outlier decisions that are difficult to justify in concrete terms due to its randomized partitioning [Mykhaylova et al. 2024, Ceschin et al. 2025]. Following this perspective, we treat explanations as meaningful only when grounded in controlled evaluation, and therefore study IF under fixed benign baselines and contamination sweeps on CICIoT2023 flood scenarios [Neto et al. 2023].

Prior work on explainability in intrusion detection has largely relied on post hoc attribution methods. [Zouhri et al. 2024] reflect the common view that interpretability often focuses on understanding black-box decisions through aggregate feature contributions. In this line, SHAP and LIME are widely adopted in IDS studies [Bacevicius et al. 2025, Gaitan-Cardenas et al. 2023, Afnan Birahim et al. 2025], but their explanations depend on perturbation, sampling, or background assumptions, which can limit stability and auditability. In a related cybersecurity setting, [Mykhaylova et al. 2024] use IF for brute-force login detection and highlight the practical importance of interpretability, but do not provide explanations. More directly, [Mahmud and Lendak 2024] use IF together with SHAP after filtering likely normal samples, while [Rachwał et al. 2024] combine SHAP-derived weights with multiple retrained IF models. Beyond IF, XAINIDS [Dasari et al. 2025] adopts a stacked ensemble with LIME and SHAP explanations. Although these approaches improve interpretive access, they remain post hoc and may introduce instability, additional computational cost, or a mismatch between what the model learns and what the explanation conveys.

Our work is closer to approaches that seek explanations aligned with the model's own decision structure. DPG [Arrighi et al. 2024, Ceschin et al. 2025] addresses this need by producing model-faithful, audit-ready explanations based on predicate relations rather than surrogate attributions alone. In our prior work [Pereira et al. 2025], we used DPG to make a *supervised* RF IDS structurally auditable through predicate-graph topology, including local reaching centrality (LRC), communities, and class bounds, going beyond feature-importance plausibility checks. Here, we transfer this auditability goal to the *unsupervised* anomaly-detection setting by building DPG on top of IF to extract traceable predicate constraints and decision-path evidence for why individual flows are flagged, complementing IF explainability efforts that primarily rely on SHAP or LIME [Bacevicius et al. 2025, Gaitan-Cardenas et al. 2023, Dasari et al. 2025].

3. Materials and Methods

3.1. Dataset

For our data, we conduct a controlled benchmark on the CICIoT2023 dataset, a large-scale flow collection with 46,686,579 instances, including 1,098,195 benign flows and tens of millions of flood-style attack flows (e.g., 33,984,560 DDoS and 8,090,738 DoS) [Neto et al. 2023]. The dataset includes heterogeneous threats beyond flooding, such as scanning and host discovery (Recon), spoofing, dictionary brute force, and web attacks (e.g., SQL injection and XSS). We focus on DoS/DDoS flood-style attacks because they are common and high-impact in IoT botnets, dominate the dataset in sample volume (enabling stable controlled sweeps and reliable scenario selection), and are strongly reflected in flow statistics, making predicate-level explanations more actionable for analysts.

Rather than manually choosing scenarios, we derive single-, dual-, and triple-attack settings from the benchmark itself. We run the benchmark over all eligible DoS/DDoS families and all two- and three-family combinations under a fixed benign protocol and shared sweeps (Section 3.2). For each candidate scenario, we aggregate performance over the grid using medians for both score-space separability (AUC and percentile gaps) and thresholded detection (F1 and FPR). We then select one representative scenario by ranking candidates by best median separability (AUC and gap), using F1 and lower FPR as tie-breakers. Under our controlled benchmark, each scenario is evaluated across a shared hyperparameter grid and contamination sweeps, which would make repeated training and testing on the full benign set computationally prohibitive (runtime and memory) and would hinder reproducibility.

Although CICIoT2023 is frequently used as a benchmark, in this study it is not treated as an abstract optimization task. The dataset is built from real packet captures of IoT devices and preserves protocol-level semantics, traffic rates, and TCP flag dynamics that are central to flood-style attacks. Our experimental design intentionally fixes benign baselines and varies training contamination and attack mixtures to reflect operational constraints faced by anomaly-based IDS deployments, such as label scarcity and non-stationary traffic. This setup allows us to analyze detection outcomes and explanation changes under controlled but realistic conditions.

In CICIoT2023, each sample is represented by a vector of 47 network features capturing various network traffic characteristics, Table 1 lists the features used in this study, their operational meaning, and the observed value ranges (min–max) reported for the dataset.

3.2. Isolation Forest implementation

This systematic selection ensures that the interpretability analysis is conducted on top-ranked scenarios whose behavior is stable across the contamination sweep, avoiding explanations of inaccurate models. The best selected scenarios are `DDOS-HTTP_FLOOD` (single), `DDOS-HTTP_FLOOD+DDOS-RSTFINFLOOD` (dual) and `DDOS-ICMP_FLOOD+DDOS-RSTFINFLOOD+DOS-HTTP_FLOOD` (triple).

To keep the study computationally tractable while preserving underlying statistics, we draw a fixed subsample of benign traffic from all available datasets and keep this benign subset constant across all scenarios and sweeps. We split benign flows into disjoint

Table 1. Subset of CICIoT2023 features explicitly cited in this paper, with operational meaning and observed value range (min–max) reported for the dataset.

Feature	Meaning (operational definition)	Range (min–max)
Rate	Packet transmission rate in the flow	0 – 8,388,608
Number	Number of packets in the flow	1 – 15
Duration	Time-to-Live (TTL)	0 – 255
Protocol Type	Protocol type (IP/UDP/TCP/IGMP/ICMP/Unknown)	0 – 47
HTTP	Application-layer protocol indicator (HTTP)	0 – 1
HTTPS	Application-layer protocol indicator (HTTPS)	0 – 1
DNS	Application-layer protocol indicator (DNS)	0 – 1
SSH	Application-layer protocol indicator (SSH)	0 – 1
IRC	Application-layer protocol indicator (IRC)	0 – 1
TCP	Transport-layer protocol indicator (TCP)	0 – 1
DHCP	Application-layer protocol indicator (DHCP)	0 – 1
ARP	Link-layer protocol indicator (ARP)	0 – 1
ICMP	Network-layer protocol indicator (ICMP)	0 – 1
Tot sum	Sum of packet lengths in the flow	42 – 127,335.8
Max	Maximum packet length in the flow	42 – 49,014
IAT	Inter-arrival time (difference from previous packet timestamp)	0 – 167,639,436
Covariance	Covariance between incoming and outgoing packet lengths	0 – 154,902,159
Variance	Variance of incoming/outgoing packet lengths in the flow	0 – 1

training and test partitions (70/30) and reuse them throughout the benchmark: *Benign-Train-Fixed* is the benign subset used for training in all runs and *Benign-Test-Fixed* is the benign subset used for testing in all runs.

For a requested training-contamination ratio r, we construct the training set by combining *Benign-Train-Fixed* with a sampled number of malicious flows A_{train} such that $A_{\text{train}}/|\textit{Benign-Train-Fixed}| \approx r$. The test set is not built to match r: it combines *Benign-Test-Fixed* with a comparable malicious set obtained by capping the total number of malicious test flows to at most $|\textit{Benign-Test-Fixed}|$ and distributing this malicious budget evenly across the attack families present in the scenario. This prevents test imbalance from dominating comparisons and isolates the effects of training contamination and scenario heterogeneity.

We train an IF anomaly detector for each scenario and training-contamination setting. For each scenario, we vary the training-contamination ratio over a non-uniform grid denser at low contamination (0.005 steps from 0.005 to 0.05, 0.01 steps from 0.06 to 0.10, and 0.05 steps from 0.15 to 0.50), and we sweep the ensemble size `n_estimators` from 5 to 100 in steps of 5.

3.3. Feature selection and evaluation metrics

As an additional experimental setting, we apply a filter-based feature-selection stage combining ANOVA F-test ranking and correlation filtering (ANOVA+CON) prior to training [Bader et al. 2026]. Filter methods are common in IDS pipelines because they are fast, scalable to high-dimensional data, and reduce computational cost by removing irrelevant or redundant variables [Bader et al. 2026]. Here, ANOVA+CON is primarily adopted to control DPG complexity: since the graph is built from IF predicates, more features generate more (often redundant) predicates and edges, which harms readability and increases computational overhead. We therefore keep only informative and non-redundant features. Because the ANOVA F-test is supervised, ANOVA+CON requires both benign and attack samples to compute class-conditional statistics. Therefore, it is only applied for contaminated training ($r > 0$); for clean training ($r = 0$), results are reported without

ANOVA+CON.

We report precision, recall, F1, and false positive rate (FPR) for thresholded detection. We also evaluate score-space separability independently of a decision threshold using AUC computed from raw anomaly scores and a percentile-based separation measure, referred to here as the interquartile gap (IQR gap), defined as `attack_p25` − `benign_p75`. To reduce noise from sampling variability across configurations, we summarize results using the median across repeated runs within each configuration.

3.4. Decision Predicate Graph analysis

DPG provides a structured, global explanation layer by converting the internal decision predicates of a tree ensemble into a single directed weighted graph [Arrighi et al. 2024]. Each node represents a predicate (a feature-threshold test) that appears in the trees, and directed edges encode predicate co-occurrence along decision paths: when a predicate p_i is followed by p_{i+1} on root-to-leaf traversals, the graph includes an edge $p_i \rightarrow p_{i+1}$, whose weight aggregates how frequently this transition occurs across trees and samples.

This representation makes the model's logic inspectable at predicate depth, enabling ranking and interpretation through graph metrics: Inlier Outlier Propagation Score (IOPS) captures whether a predicate predominantly propagates toward the outlier outcome (more negative implies stronger outlier-driving behavior), LRC measures how much of the graph is reachable from a predicate (how broadly it conditions downstream structure) [Mones et al. 2012], betweenness highlights predicates that bridge distinct regions of the graph [Brandes 2008], and community structure is obtained by first extracting a weighted backbone of the DPG (retaining the strongest co-occurrence links to avoid a single dense component) and then applying label-propagation clustering, so that each community groups predicates that frequently co-activate along isolation paths into recurring decision categories.

To interpret the resulting communities, we use the sign of IOPS as a behavioral cue: in our anomaly-detection formulation, predicates with IOPS> 0 tend to align with the inlier side, whereas predicates with IOPS< 0 tend to align with the outlier side. For each community C, we compute its mean IOPS, $\overline{\mathrm{IOPS}}(C)$, and label it benign-centric when $\overline{\mathrm{IOPS}}(C) > 0$ and outlier-centric when $\overline{\mathrm{IOPS}}(C) < 0$. In each scenario, we contrast these modules and report the most negative-IOPS predicates inside the outlier-centric module as the main outlier drivers. To contextualize DPG explanations, we include two widely used post-hoc interpretability baselines. SHAP provides attribution scores by relating a model's prediction to Shapley values from cooperative game theory, yielding a principled feature-contribution view that can be aggregated into global importance summaries [Lundberg and Lee 2017]. LIME explains individual predictions by fitting a simple, local surrogate model around a target instance, producing instance-level explanations in terms of locally influential features [Ribeiro et al. 2016]. We compute these baselines on the same feature space and detector output, enabling direct comparison with the predicate and co-occurrence structure exposed by DPG.

The asymptotic complexity of DPG construction is $O\big(b \times s \times (k + k^2)\big)$, where b is the number of learners in the ensemble, s is the number of samples in the training set, and k denotes the size of the predicate and edge structures (P_x, E_x) processed by the TRAVERSING and AGGREGATING functions [Arrighi et al. 2024].

3.5. Code availability and Use of Generative AI

We build upon the original DPG-iForest reference implementation[1]. Our implementation, experiment scripts, and configuration files are available in our repository[2]. All experiments were conducted in Python 3.12.11 using scikit-learn 1.5.1, NumPy 2.0.0, pandas 2.2.2, NetworkX 3.3, Matplotlib 3.9.1, and joblib 1.4.2.

A generative AI tool (ChatGPT, OpenAI) was used to review, transcribe, and paraphrase portions of the manuscript to improve clarity and fluency. All outputs were reviewed by the authors, who take full responsibility for the entire content.

4. Results and Discussion

4.1. RQ1: Attack fraction injected in training and scenario effects

Figure 1 summarizes how attack fraction reshapes IF performance. With fully clean training (only benign), the detector produces virtually no false positives (Figure 1b) but fails to flag a large portion of malicious traffic, which keeps F1 low despite high precision (Figure 1a).

Introducing a small fraction of malicious traffic during training shifts the operating point to a better trade-off (Figure 1). In the low-contamination regime (0.005%–3%), F1 rises sharply while FPR remains near the floor (Figure 1a–b). With ANOVA+CON, the best region occurs at slightly lower contamination, suggesting feature selection strengthens benign–attack contrast when attacks are still rare, improving detection without increasing false positives (Figure 1a–b). Since ANOVA is supervised, ANOVA+CON is undefined for $r = 0\%$ and is applied only for $r > 0\%$; clean-training results are therefore reported without ANOVA+CON. Beyond this regime, performance degrades as contamination increases, with FPR growing (Figure 1b) and score separability collapsing (Figure 1c), as reflected by shrinking AUC and percentile gaps. This reflects a normalization effect in which frequent attacks in training are partially absorbed into the learned baseline, bringing benign and attack score distributions closer and worsening thresholded classification. It is important to note that, in real Security Operations Centers (SOC), contamination is not a directly controllable quantity, especially during large-scale flood events in which malicious traffic may suddenly dominate the network. Therefore, the contamination sweep in this study was not intended to represent an operational target that practitioners can explicitly enforce, but rather to identify, under controlled conditions, the regime in which the detector behaves best. In this sense, the low-contamination range should be interpreted as an analytical reference for understanding model behavior, rather than as a directly tunable parameter in practice.

Figure 2 shows the same effect in score space by plotting IF anomaly-score percentile bands for benign and attack test samples. For each contamination setting, we compute scores for each group and summarize them with the $p25$–$p75$ band and the median. At low contamination, the bands are essentially disjoint and the interquartile gap reaches its maximum, indicating clear separation near the inlier boundary. As contamination increases, the bands converge and the gap shrinks, eventually becoming negative, meaning a non-trivial fraction of attacks overlaps the upper tail of benign scores and blurs

[1] `https://github.com/Math0097/DPG-iForest`
[2] `https://github.com/Eronponce/IF-DPG-SBRC`

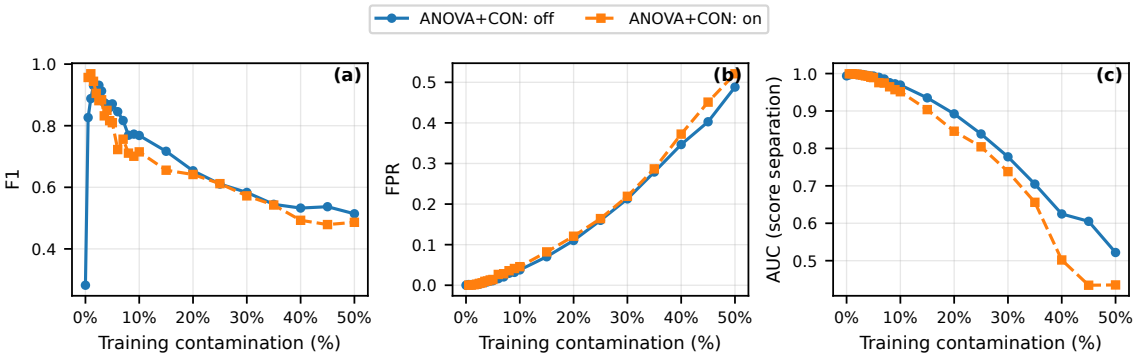

Figure 1. Overall medians versus training contamination for F1, FPR, and anomaly-score AUC, comparing ANOVA+CON on/off.

the decision boundary. This trend mirrors the AUC drop in Figure 1c and explains the increase in false positives and the reduction in F1.

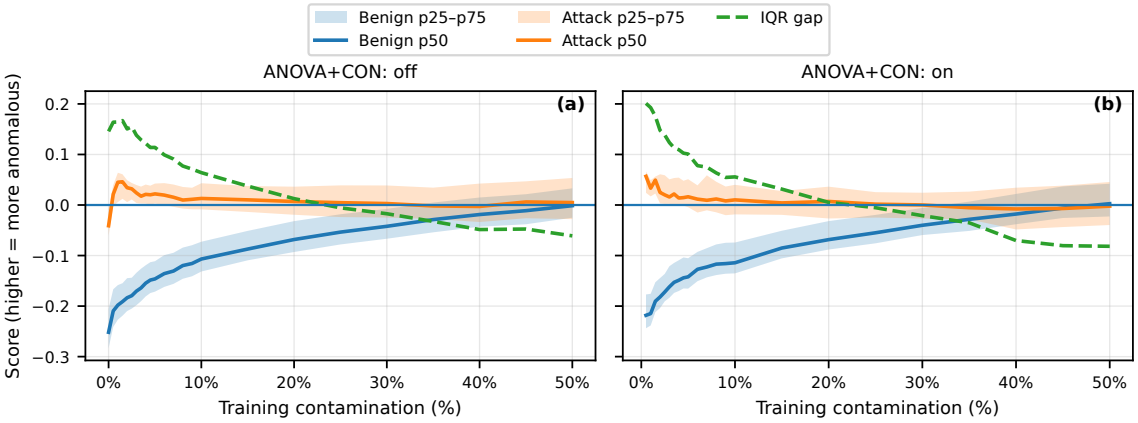

Figure 2. Score-space analysis: benign and attack percentile bands (p25–p75) and medians, plus the interquartile gap (attack p25 minus benign p75), versus training contamination.

Across scenarios, the best operating region occurs at low contamination (0.005%–3%). As contamination increases, the interquartile gap decreases consistently across scenarios, most notably for the dual-attack setting, signaling growing overlap between benign and attack score distributions and mirrored declines in F1/recall (Figure 3a–c). Beyond this interval, the normalization failure mode dominates and AUC drops (Figure 1c), percentile bands overlap strongly, and the gap becomes negative (Figure 2), indicating that attacks cease to appear "rare" and are increasingly absorbed into the model's notion of normality.

We can now consolidate the main findings by answering the research question RQ1 as follows. For RQ1, purely benign training keeps false positives very low but reduces sensitivity to flood attacks, whereas introducing a small amount of contamination improves score-space separability and produces the best recall–FPR trade-off, while higher contamination progressively harms discrimination as attack patterns become partially absorbed as normal behavior.

4.2. RQ2: DPG semantic validity

We interpret the trained IF models through their DPG, focusing on (i) IOPS, (ii) coupled predicate patterns via co-occurrence edges, (iii) bridging predicates via betweenness, and

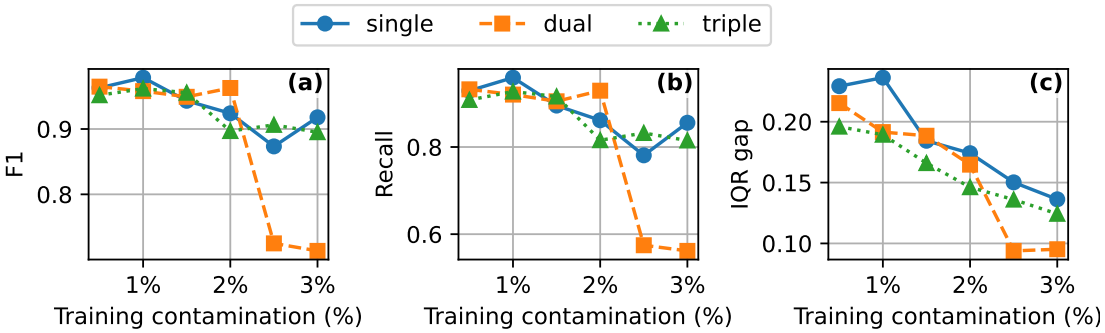

Figure 3. Low-contamination regime (ANOVA+CON on, n_estimators=50): median (a) F1, (b) recall, and (c) score separability (IQR gap) versus training contamination.

(iv) community structure.

In the single-attack scenario, outlier-driving predicates concentrate on TCP activity counters and flag-related fields. Table 2a lists the top predicates by most-negative IOPS. Notably, `syn_count >`, `rst_count >`, and `fin_count >` dominate the outlier propagation signal, consistent with volumetric TCP connection churn and abnormal flag distributions under flood-style behavior.

Table 2. Single-attack scenario: outlier-driving predicates and dominant co-occurrences.

(a) Top outlier using IOPS.

Predicate	IOPS
syn_count >	-0.812
rst_count >	-0.552
Number >	-0.520
fin_count >	-0.405
rst_flag_number >	-0.399

(b) Top outlier transitions (weighted frequency).

From	To	Wfreq
fin_count >	Number >	8194.397
rst_count >	rst_count >	7141.749
rst_flag_number >	rst_flag_number >	4605.463
Number >	fin_count >	2400.085
rst_count >	rst_flag_number >	1413.012

Although the CICIoT2023 `DDOS-HTTP_FLOOD` traces are generated with an application-layer HTTP flooding tool (`golang-httpflood`) [Neto et al. 2023, Leeon123 2020], the DPG still highlights TCP-level counters and flag aggregates, which is consistent with the connection patterns induced by high-rate HTTP flooding.

Co-occurrence edges among the most-negative-IOPS predicates indicate that the DPG explanation is characterized by coupled predicate pairs rather than a single isolated counter (Table 2b). In particular, the bidirectional high-weight link `fin_count` $> \tau \rightarrow$ `Number` $> \tau$ and `Number` $> \tau \rightarrow$ `fin_count` $> \tau$ suggests that elevated FIN activity co-occurs with elevated overall traffic volume along outlier decision paths. Likewise, the strong edge `rst_count` $> \tau \rightarrow$ `rst_flag_number` $> \tau$ links a TCP counter with its flag aggregate, consistent with bursty connection teardown/reset behavior under high-rate flooding. Self-loops (e.g., `rst_count` $> \rightarrow$ `rst_count` $>$) reflect repeated reuse of the same predicate label along outlier paths and should be interpreted as persistence of a dominant condition rather than a cross-feature interaction.

In the dual-attack Scenario, the outlier-driving set remains largely consistent with

the single-attack case, continuing to emphasize transport-level counters and generic intensity signals rather than introducing a clearly attack-specific signature. Table 3a lists the top predicates by most-negative IOPS. The strongest signals are `rst_count >` (IOPS = -0.664) and `syn_count >` (IOPS = -0.591), followed by `Number >` (IOPS = -0.425), `cwr_flag_number >` (IOPS = -0.373), and `syn_flag_number >` (IOPS = -0.347). This overlap should not be read as evidence that the IF distinguishes one attack from the other; rather, it suggests that the model relies on a stable *shared flood footprint* to separate benign traffic from a broader class of volumetric DoS/DDoS behavior.

Table 3. Dual-attack scenario: outlier-driving predicates and dominant co-occurrences.

(a) Top outlier using IOPS.

Predicate	IOPS
rst_count >	-0.664
syn_count >	-0.591
Number >	-0.425
cwr_flag_number >	-0.373
syn_flag_number >	-0.347

(b) Top outlier transitions (weighted frequency).

From	To	Wfreq
rst_count >	Rate >	7593.871
syn_flag_number >	syn_count >	6515.970
syn_count >	syn_flag_number >	5128.211
Number >	syn_flag_number >	4497.381
Rate >	rst_count >	3498.188

This interpretation is consistent with CICIoT2023 flow features, which are computed as mean statistics over fixed packet windows (10 and 100 packets), emphasizing aggregate flood indicators over fine-grained attack cues [Neto et al. 2023]. It also matches the dataset tooling: although RST/FIN floods are packet-crafted (e.g., `hping3`) and HTTP floods are application-layer, both can induce elevated transport-layer reset/counter patterns under overload, making `rst_count` a plausible cross-attack mediator [Tools 2019, Leeon123 2020, Neto et al. 2023]. DPG supports this shared logic by showing stable outlier-driving predicates and co-occurrence structure when moving from single to dual mixtures.

The dominant couplings make the mixture effect explicit (Table 3b): `rst_count>` $\leftrightarrow$ `Rate>` captures the intensity–reset linkage, `syn_flag_number>` $\leftrightarrow$ `syn_count>` captures SYN dynamics, and `Number>` $\rightarrow$ `syn_flag_number>` links volume to abnormal flag patterns. Overall, DPG recovers the expected coupling between high-rate traffic and abnormal TCP flag behavior in blended flood scenarios.

In the triple-attack scenario, DPG highlights ICMP participation, intensity, and TCP control-plane predicates. The strongest outlier drivers are `rst_count>`, `ICMP>`, and `Number>`, followed by `cwr_flag_number>` and `Rate>` (Table 4a). This points to an ICMP-heavy, high-throughput regime with abnormal TCP flag/counter behavior, consistent with the mixture composition [Tools 2019, Leeon123 2020, Neto et al. 2023]. The dominant couplings (Table 4b), including `rst_count>`$\rightarrow$`Rate>` and `Number>`$\rightarrow$`ICMP>`, reinforce this separation.

4.2.1. Community structure

We report community structure using asynchronous Label Propagation (LPA) on the backbone-filtered undirected DPG, following [Arrighi et al. 2024]. Across scenarios,

Table 4. Triple-attack scenario: outlier-driving predicates and dominant co-occurrences.

(a) Top outlier using IOPS.

Predicate	IOPS
rst_count >	-0.448
ICMP >	-0.379
Number >	-0.348
cwr_flag_number >	-0.271
Rate >	-0.200

(b) Top outlier transitions (weighted frequency).

From	To	Wfreq
rst_count >	Rate >	7745.306
Rate >	Rate >	7744.947
ICMP >	ICMP >	4714.473
syn_flag_number >	syn_flag_number >	2350.671
Number >	ICMP >	1913.261

LPA consistently separates a large benign-centric module (positive mean IOPS) from a compact outlier-centric module (negative mean IOPS); we summarize the main outlier drivers as the most negative-IOPS predicates in the outlier-centric module.

In the single-attack scenario, the outlier-centric module is driven by `syn_count>`, `rst_count>`, `Number>`, `fin_count>`, and `rst_flag_number>`. The dual-attack scenario shows a similar outlier module centered on `rst_count>`, `syn_count>`, `Number>`, `syn_flag_number>`, and `Rate>`, while the benign module is dominated by complementary "$\leq$" predicates (e.g., `Duration`$\leq$, `IAT`$\leq$). In the triple-attack scenario, LPA yields an additional small residual module and an outlier-centric module highlighted by `rst_count>`, `ICMP>`, `Number>`, `Rate>`, and `syn_count>`, consistent with the ICMP component in the mixture.

Overall, these results answer RQ2: DPG emphasizes volumetric and TCP/ICMP predicate families and shows that alerts are driven by coupled conditions, yielding compact, traceable constraints aligned with flood semantics.

4.3. RQ3: Interpretability and detection trade-off

Alongside DPG, we report SHAP and LIME baselines to contextualize the explanation patterns. Table 5 compares SHAP (mean $|\mathrm{SHAP}|$) with the DPG-IOPS ranking per scenario. In the *single-attack* setting, SHAP mainly highlights protocol-indicator variables (e.g., `HTTPS`, `Protocol Type`, `DNS`, `HTTP`, `ARP`, `ICMP`), whereas in the *dual* and *triple-attack* settings, it shifts toward intensity and TCP flag/counter statistics (e.g., `Rate`, `ack_flag_number`, `fin_flag_number`, `rst_count`, `syn_count`). DPG-IOPS is consistent with this trend, repeatedly surfacing counter/flag predicates (e.g., `syn_count`, `rst_count`, `fin_count`) together with volume variables (e.g., `Number`, `Rate`). Both baselines and DPG indicate that anomaly decisions are driven by protocol presence and volume/flag dynamics across scenarios.

LIME produces sparse, instance-level rules by fitting a local linear surrogate per flow; we summarize these rules by counting how often they appear with positive (pro-anomaly) or negative (anti-anomaly) weight. Table 6 contrasts recurrent LIME rules with the top DPG predicates ranked by $|\mathrm{IOPS}|$ (direction preserved). In the single and dual scenarios, both emphasize TCP counter regimes (e.g., `rst_count`, `syn_count`) as pro-anomaly evidence, while differences appear mainly in counter-evidence: LIME often relies on protocol-indicator presence/absence, whereas DPG highlights stable inlier constraints learned by the ensemble. In the triple-attack scenario, the methods may select different but correlated signatures, reflecting local surrogate variability versus DPG's

Table 5. Global attribution: SHAP vs. DPG-IOPS. Top-8 per scenario (rank only). DPG-IOPS is shown as predicate direction.

	Single	Dual	Triple
SHAP	HTTPS Protocol Type DNS HTTP ARP syn_flag_number ICMP Max	HTTPS HTTP Rate fin_flag_number ack_flag_number Tot sum rst_count syn_count	HTTPS Rate fin_count fin_flag_number TCP ack_flag_number rst_count Header_Length
DPG-IOPS	syn_count $> \tau$ rst_count $> \tau$ IRC $> \tau$ Number $> \tau$ cwr_flag_number $\leq \tau$ cwr_flag_number $> \tau$ fin_count $> \tau$ rst_flag_number $> \tau$	rst_count $> \tau$ IRC $> \tau$ syn_count $> \tau$ Number $> \tau$ cwr_flag_number $> \tau$ syn_flag_number $> \tau$ ece_flag_number $> \tau$ Rate $> \tau$	rst_count $> \tau$ ICMP $> \tau$ Number $> \tau$ cwr_flag_number $> \tau$ SSH $> \tau$ Rate $> \tau$ IGMP $> \tau$ syn_flag_number $> \tau$

recurrence along outlier paths.

Table 6. DPG vs. LIME (label=anomaly): compact comparison by scenario. DPG predicates are ranked by $|\mathrm{IOPS}|$ (negative = outlier evidence; positive = inlier evidence). Thresholds are denoted by τ.

Scenario	Summary (rank only)
Single	**DPG (outlier):** syn_count $> \tau$; rst_count $> \tau$ **LIME (pro-anomaly):** rst_count > 0; Rate > 0 **DPG (inlier):** IRC $> \tau$; cwr_flag_number $\leq \tau$ **LIME (counter):** low/zero ICMP/DHCP/SSH indicators; Protocol Type
Dual	**DPG (outlier):** rst_count $> \tau$; syn_count $> \tau$ **LIME (pro-anomaly):** syn_count > 0; syn_flag_number > 0 **DPG (inlier):** IRC $> \tau$; ece_flag_number $> \tau$ **LIME (counter):** HTTP/HTTPS indicators; low/zero DHCP/IGMP indicators
Triple	**DPG (outlier):** rst_count $> \tau$; ICMP $> \tau$ **LIME (pro-anomaly):** Variance > 0; syn_count > 0 **DPG (inlier):** SSH $> \tau$; IGMP $> \tau$ **LIME (counter):** low/zero HTTP indicator; low syn_flag_number/syn_count

SHAP and LIME provide global and local attributions but do not capture how predicate conditions interact along decision paths. DPG complements them by working in predicate space and exposing co-occurrence and graph structure (e.g., betweenness), enabling constraint- and path-based summaries of the detector's logic. While SHAP/LIME and DPG highlight overlapping feature families, DPG adds interaction and decision-structure evidence by extracting predicate constraints and their co-occurrence directly from the isolation trees, addressing RQ3.

5. Conclusion

This paper studied how to make anomaly-based IoT intrusion detection more operationally transparent by interpreting trained IF models with DPGs, which expose feature-threshold predicates and their co-occurrence structure without altering the detector's scoring function. In a controlled benchmark on CICIoT2023 flood scenarios, including single-, dual-, and triple-attack mixtures, we found that performance is maximized in a low-contamination regime, where introducing a small fraction of malicious traffic during training improves score-space separability and thresholded detection. Across scenarios, DPG

rankings and edges consistently highlighted volumetric and TCP or ICMP dynamics and showed that outlier decisions are driven by coupled predicate patterns. Although this study relies on fixed benign baselines, real deployments are more dynamic and may include stealthier attacks that resemble legitimate traffic, which can reduce discrimination by making malicious patterns appear normal to the detector. In this context, explainability can play a practical role beyond alert inspection, since DPG-based evidence may help reveal unstable predicates and decision structures, supporting model revision and improving how the detector is maintained over time. Future work will produce per-flow, compact predicate summaries and evaluate their impact on analyst triage and false-positive reduction, as well as investigate how DPG-derived metrics can be integrated into textual dashboards and actionable analyst rules, reducing the need to visually inspect the full graph during time-sensitive incident handling, while also assessing scalability and drift in more diverse network environments.

References

Afnan Birahim, S., Paul, A., Rahman, F., Islam, Y., Roy, T., Asif Hasan, M., Haque, F., and Chowdhury, M. E. H. (2025). Intrusion detection for wireless sensor network using particle swarm optimization based explainable ensemble machine learning approach. *IEEE Access*, 13:13711–13730.

Arrighi, L., Pennella, L., Marques Tavares, G., and Barbon Junior, S. (2024). Decision predicate graphs: Enhancing interpretability in tree ensembles. In *World Conference on Explainable Artificial Intelligence*, pages 311–332. Springer.

Bacevicius, M., Paulauskaite-Taraseviciene, A., Zokaityte, G., Kersys, L., and Moleikaityte, A. (2025). Comparative analysis of perturbation techniques in lime for intrusion detection enhancement. *Machine Learning and Knowledge Extraction*, 7(1).

Bader, A., Salim, O., Khudhur, O., Al-Barzinji, S., and Jasem, F. (2026). Feature selection techniques in intrusion detection systems: A review. *Journal of Cybersecurity and Information Management*, 17:97–112.

Brandes, U. (2008). On variants of shortest-path betweenness centrality and their generic computation. *Social Networks*, 30(2):136–145.

Ceschin, M., Arrighi, L., Longo, L., and Barbon Junior, S. (2025). Extending decision predicate graphs for comprehensive explanation of isolation forest. In *World Conference on Explainable Artificial Intelligence*, pages 271–293. Springer.

Dasari, A., Bisawas, S., and Purkayastha, B. (2025). Enhanced network intrusion detection systems with explainable artificial intelligence for network security. *International Journal of Communication Systems*, 38(14):e70209. e70209 IJCS-25-1913.R1.

Gaitan-Cardenas, M. C., Abdelsalam, M., and Roy, K. (2023). Explainable ai-based intrusion detection systems for cloud and iot. In *2023 32nd International Conference on Computer Communications and Networks (ICCCN)*, pages 1–7.

Gunning, D., Vorm, E., Wang, J. Y., and Turek, M. (2021). Darpa's explainable ai (xai) program: A retrospective. *Applied AI Letters*, 2(4):e61.

Héder, M. (2023). Explainable ai: A brief history of the concept. *ERCIM NEWS*, (134):9–10.

Johnstone, J. and Akinfaderin, A. (2025). Mapping cyber threats in iot-driven msps: An explainable machine learning approach for remote work security. In *2025 IEEE 4th International Conference on AI in Cybersecurity (ICAIC)*, pages 1–9.

Leeon123 (2020). Golang-httpflood. `https://github.com/Leeon123/golang-httpflood`. GitHub repository. Accessed: 2025-01-20.

Liu, F. T., Ting, K. M., and Zhou, Z.-H. (2008). Isolation forest. In *2008 Eighth IEEE International Conference on Data Mining*, pages 413–422.

Lundberg, S. and Lee, S.-I. (2017). A unified approach to interpreting model predictions.

Mahmud, J. S. and Lendak, I. (2024). Enhancing one-class anomaly detection in unlabeled datasets through unsupervised data refinement. In *2024 IEEE 22nd Jubilee International Symposium on Intelligent Systems and Informatics (SISY)*, pages 000497–000502.

Mones, E., Vicsek, L., and Vicsek, T. (2012). Hierarchy measure for complex networks. *PLOS ONE*, 7(3):1–10.

Mutambik, I. et al. (2024). An efficient flow-based anomaly detection system for enhanced security in iot networks. *Sensors*, 24(22):7408.

Mykhaylova, O., Shtypka, A., and Fedynyshyn, T. (2024). An isolation forest-based approach for brute force attack detection. In *BAIT'2024: The 1st International Workshop on Bioinformatics and applied information technologies*, volume 3842, pages 43–54, Zboriv, Ukraine. CEUR Workshop Proceedings.

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(13).

Nwakanma, C. I., Ahakonye, L. A. C., Jun, T., Lee, J. M., and Kim, D.-S. (2023). Explainable scada-edge network intrusion detection system: Tree-lime approach. In *2023 IEEE International Conference on Communications, Control, and Computing Technologies for Smart Grids (SmartGridComm)*, pages 1–7.

Pereira, E. P., Moradbeikie, A., Zarpelão, B. B., and Barbon Junior, S. (2025). Learning to explain cyberattacks: Insights from random forest and decision predicate graphs. In *Proceedings of the Thematic Workshops at Ital-IA 2025*, volume 4121 of *CEUR Workshop Proceedings*.

Rachwał, A., Karczmarek, P., Rachwał, A., and Stegierski, R. (2024). Isolation forest with exclusion of attributes based on shapley index. *IEEE Access*, 12:101797–101813.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). "why should i trust you?": Explaining the predictions of any classifier.

Tools, K. (2019). hping3 package description. `https://tools.kali.org/information-gathering/hping3`. Accessed: 2026-01-23.

Zouhri, H., Idri, A., and Hakkoum, H. (2024). Assessing the effectiveness of dimensionality reduction on the interpretability of opaque machine learning-based attack detection systems. *Computers and Electrical Engineering*, 120:109627.