BUCKINGHAMSHIRE
NEW UNIVERSITY
EST. 1891

Downloaded from: https://bnu.repository.guildhe.ac.uk/

This document is made available with permission. Copyright remains with the author(s).

Use of items from Buckinghamshire New University's institutional repository must comply with the usage guidelines.

This work is made available under a **Creative Commons Attribution-4.0 International Licence (CC BY 4.0)**: https://creativecommons.org/licenses/by/4.0/

Under this licence, you may:
• Share — copy and redistribute the material in any medium or format
• Adapt — remix, transform, and build upon the material for any purpose, even commercially.

You must:
• cite the authors, title, and full bibliographic details of the item
• include a hyperlink/URL to the repository record in any citation
• keep all files required for use of the item together
• Indicate of changes were made.

You may not:
• sell any part of an item
• refer to any part of an item without citation
• use or adapt the work in a way that misrepresents the author or impugns their reputation
• remove or alter the copyright statement on an item
• You may not apply legal terms or technological measures that legally restrict others from doing anything the license permits.

If you need further guidance contact the Library and Learning Services at Library@bnu.ac.uk

electronics

MDPI

Article

Leakage-Controlled Attack Classification with Cyber Kill Chain Stage Mapping for IoT Devices in SOHO Environments

Benjamin Aziz *, Estabraq H. Makiyah and Ukamaka Oragwu

College of Creative Arts, Technology and Engineering, Buckinghamshire New University, High Wycombe HP11 2JZ, UK; estabraq.makiyah@bnu.ac.uk (E.H.M.); ukamaka.oragwu@bnu.ac.uk (U.O.)
* Correspondence: benjamin.aziz@bnu.ac.uk

Abstract

Machine-learning (ML) intrusion detection studies on the CICIoT2023 IoT dataset routinely report near-perfect accuracy. We show that, for the widely used class-balanced six-class release, such figures are an artefact of the evaluation protocol. A data audit finds that 39.0% of the 1.8 million records are exact duplicates introduced by oversampling: under the conventional random 80/20 split, 48.6% of test records have an identical copy in the training set. We also find that the inter-arrival-time feature behaves as a capture-time stamp, achieving 87.4% accuracy on its own. Under the conventional protocol, Random Forest reaches 99.86% accuracy. With duplicates removed, the shortcut feature excluded, capture-time-blocked five-fold cross-validation and nested hyperparameter tuning, the best of seven models (XGBoost) reaches a macro-F1 of 0.733 (95% CI ± 0.098), and the three tree ensembles are statistically indistinguishable. At a realistic 1% attack prevalence, alert precision falls to 8.3%. On an independently processed, duplicate-free release of the full CICIoT2023 capture, holding out entire capture files lowers macro-F1 from 0.803 under a random split to 0.720, and the confusions between DDoS and DoS and among benign, reconnaissance and web-attack traffic recur. Each attack category is mapped to a single Cyber Kill Chain (CKC) stage, so the stage labels are derived deterministically from the attack categories and no new stage classifier is proposed. Instead, we evaluate what reporting results at stage level adds beyond an attack label: merging the DDoS and DoS categories into a single Delivery stage removes a confusion that the flow features of both releases cannot resolve; a previously unseen Delivery subtype is still assigned to the Delivery stage; and cost-sensitive stage decisions reduce missed attacks. Stage labels are assigned at category level, so the analysis does not infer an attack's progression through the kill chain over time. SHAP analysis is repeated on the leakage-controlled model. We release the audit and the evaluation code so that results on this dataset can be reported reproducibly.

Keywords: intrusion detection system; Internet of Things; small office small home networks; Cyber Kill Chain; machine learning; explainable AI; data leakage; evaluation methodology

Check for updates

Academic Editors: Zbigniew Kotulski and Aryya Gangopadhyay

Received: 4 September 2026
Revised: 10 November 2026
Accepted: 17 November 2026
Published: 20 November 2026

Copyright: © 2026 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license.

1. Introduction

The Internet of Things (IoT) has become integral to Small Office and Home Office (SOHO) environments, where devices such as smart thermostats, microphones, cameras, and voice assistants now sit alongside conventional computing equipment. Global IoT deployment exceeded 14 billion devices in 2024 and is projected to surpass 25 billion by 2030 [1], and this growth has been matched by expansion of the attack surface available to adversaries. IoT ecosystems are particularly exposed because devices vary widely in

protocols and vendor standards, frequently ship with weak default credentials, receive infrequent patching and lack the processing headroom needed for conventional endpoint protection [2]. SOHO networks further compound this exposure: unlike enterprise environments, they typically lack layered defences or dedicated monitoring, making them disproportionately attractive targets. Once compromised, such devices are readily conscripted into large-scale botnets as demonstrated by the Mirai campaign that disrupted major internet services in 2016 [3].

These risks are not hypothetical. In 2024, the UK National Cyber Security Centre and international partners issued a joint advisory describing how a China state-sponsored threat group had exploited outdated, unsupported SOHO devices as a launching pad for attacks against Australian networks, precisely because such devices are easier to compromise and better able to conceal malicious traffic [4]. In 2025, threat hunters uncovered a network of over 1000 compromised SOHO devices supporting a sustained Cyber-espionage campaign, LapDogs, affecting victims across the United States and Southeast Asia [5]. Separately, a destructive attack in 2024 disabled more than 600,000 SOHO routers, cutting Internet access for customers across 18 U.S. states [6]. Collectively, these incidents illustrate that SOHO IoT compromises carry consequences well beyond the individual household or business, extending into regional connectivity, critical services and international security.

ML techniques have emerged as a promising alternative to signature- and rule-based Intrusion Detection Systems (IDSs), offering the ability to detect anomalous or novel traffic patterns without relying on known attack signatures [7]. Numerous studies have applied ML algorithms to IoT intrusion detection with strong results, including on the CICIoT2023 dataset used in this study (see Section 2). However, the overwhelming majority of this work stops at classifying traffic as benign or as one of several attack types. This leaves a practically important gap: Knowing that traffic is malicious does not, by itself, indicate where in an attack's lifecycle a defender is intervening, information that is critical for prioritising response. The CKC framework [8], which decomposes an attack into sequential stages from reconnaissance through actions on objectives, offers a natural structure for this kind of stage-aware contextualisation. Yet, its application to IoT-specific traffic and to SOHO environments in particular remains largely unexplored [9]. A second gap concerns evaluation: data leakage and related pitfalls are widespread in machine-learning-based security research [10,11], and flaws in benchmark intrusion detection datasets can distort reported performance [12], so near-perfect results warrant scrutiny of how they were obtained.

1.1. Research Objectives and Contribution

This study addresses the above gap by developing a two-phase ML-based IDS for SOHO IoT traffic that (i) detects and classifies intrusions using the CICIoT2023 six-class dataset [13], and (ii) adds a CKC stage-level output, in which each detected attack category is reported as the kill-chain stage it is conventionally associated with, and evaluates what this output adds beyond the attack category. The stage is assigned at category level and is not inferred from the progression of an attack over time (Section 3.3.3). SHAP-based explainability keeps the resulting model's decisions interpretable for defenders. In doing so, the study makes four contributions:

- A data audit of the class-balanced CICIoT2023 six-class release, quantifying exact and near-duplicate records, train–test overlap under random splitting, and a capture-time shortcut feature (IAT), and showing how each inflates reported performance.
- A leakage-controlled benchmark of seven classifiers (stratified dummy, logistic regression, MLP, Decision Tree, Random Forest, LightGBM, XGBoost) using deduplicated data, capture-time-blocked five-fold cross-validation and nested randomised hyperpa-

rameter search, with confidence intervals, corrected significance tests, noise-robustness and prior-shift analyses.
- An evaluation of what a Cyber Kill Chain stage-level output adds to attack classification when the stage is derived deterministically from the attack category: a merged Delivery stage, a hierarchical detector–stage cascade, generalisation to a held-out Delivery subtype, leave-one-attack-family-out detection, and cost-sensitive stage decisions.
- SHAP explanations of the leakage-controlled model, with sample sizes and random seeds reported for replication.

1.2. Structure of the Paper

The rest of the paper is structured as follows. In Section 2, we review some of the state-of-the-art literature most relevant to our paper. In Section 3, we give details of the dataset, preprocessing and modelling methodology. In Section 4, we report the data leakage audit, the leakage-controlled and external Phase 1 intrusion detection results, the Phase 2 stage-level results and the SHAP analysis. In Section 5, we interpret these results in light of the existing literature and consider their theoretical, practical and policy relevance. Finally, in Section 6, we conclude the paper summarising the study's limitations and directions for future work.

2. Related Work

Cybersecurity research has long emphasised robust attack detection, and while SOHO environments attract less attention than higher-stakes sectors such as healthcare or critical infrastructure, the incidents outlined in Section 1 show that the consequences of SOHO compromise are far from trivial. Because SOHO-specific ML-IDS research remains limited, this review draws on three related bodies of work: studies using the CICIoT2023 dataset [13] specifically, broader ML-based IoT security research using other datasets and finally, prior work applying machine learning to CKC classification.

2.1. Machine-Learning-Based Intrusion Detection Using CICIoT2023

Several studies have used CICIoT2023 to build IoT-specific intrusion detection models. For example, in [14] the authors combine two-step data clustering with a Multilayer Perceptron to manage the dataset's high dimensionality, achieving 97.26% binary-classification accuracy. In [15], the authors applied hierarchical feature selection to reduce the feature space to 23 variables before training a CatBoost classifier, achieving 99.4% accuracy, precision, recall and F1-score across DoS, DDoS and Mirai subtypes with SHAP used to interpret feature contributions, an approach this study extends to CKC-stage classification. In [16], a comparison among five classifiers was given under binary, grouped and fine-grained labelling schemes, with Decision Tree and Random Forest achieving near-100% accuracy and substantially outperforming SVM, results that, as our audit shows (Section 4.1), can arise from record-level splitting rather than generalisation. In [17], the authors proposed a stacking ensemble (Random Forest and Linear SVM base learners with a Logistic Regression meta-model) combined with SMOTE for class balancing, achieving 93% accuracy, though with reduced precision (85%) for reconnaissance traffic due to feature overlap. This echoes the difficulty with reconnaissance traffic that we also observe under leakage-controlled evaluation. In [13], which also introduced the CICIoT2023 dataset, the authors reported 99.7% and 99.4% binary-classification accuracy for Random Forest and Deep Neural Networks respectively, but noted a marked drop in F1-score (approximately 71%) once classification was extended to the dataset's full 34-class taxonomy, illustrating the general trade-off between classification granularity and accuracy that recurs across this literature.

2.2. Broader Machine Learning-Based IoT Security Techniques

Beyond CICIoT2023 [13], a wider body of work has evaluated ML and deep learning models for IoT and industrial IoT (IIoT) intrusion detection. For example, [18] compared XGBoost, SVM and deep convolutional neural networks across IoT-23, NSL-KDD, TON_IoT and a custom testbed, finding that XGBoost achieved the best accuracy (99.98%) while training substantially faster than the alternatives, which is an efficiency argument one can use relevantly to resource-constrained SOHO deployment. In [19], the authors proposed a dual-layer LSTM architecture with cryptography-integrated alerting, achieving 98% accuracy on UNSW-NB15, while in [20], the authors found Random Forest outperforming SVM and XGBoost for both binary (92.73%) and multiclass (82.67%) classification on the same dataset, with multiclass accuracy degrading for rare attack types. This mirrors the weakness on low-volume categories (Reconnaissance and Web attack) observed in our results. In [21], an evaluation of ensemble techniques was provided within a fog-computing architecture across UNSW-NB15 and CICIDS2017, which found that stacking outperformed individual classifiers, including Random Forest though at the cost of increased model complexity and training time. In [22], the authors reported that deep learning architectures (Bi-LSTM, GRU) achieved near-perfect accuracy (99.99%) on an Industrial IoT dataset, outperforming classical ML methods but at considerably higher computational cost up to 1207 s for multiclass Bi-LSTM training. In [23], ensemble ML was combined with software-defined networking and network function virtualisation to deliver 99.9% accuracy with a 0.1% false-positive rates, demonstrating that ML-based detection can be embedded within broader automated network defence architectures. Across this literature, tree-based ensembles consistently perform competitively with, and are often more efficient than, deep learning alternatives, thus supporting the choice of tree-based ensembles (Random Forest, LightGBM and XGBoost), alongside a Decision Tree and three baselines, as the Phase 1 classifiers in our study.

2.3. Machine Learning-Based Cyber Kill Chain Classification

Literature explicitly combining machine learning with Cyber Kill Chain classification is scarce. The most directly relevant study is [24], which proposes an ML framework for mapping security alerts to the seven canonical CKC stages in the context of Advanced Persistent Threats (APTs). Using Multiple Factor Analysis for feature extraction and OneR/InfoGain for feature selection, they found that Naive Bayes achieved 91.1% accuracy using only five optimised features, outperforming SVM and Random Forest on a dataset of 3676 observations. While this confirms the general feasibility of combining CKC classification with machine learning, the work targets APT alerts rather than IoT network traffic. There are no CKC-aware classifiers proposed and trained specifically on IoT traffic, and certainly none for SOHO-relevant datasets such as CICIoT2023.

2.4. Summary and Research Gap

Table 1 summarises the reviewed literature against three criteria: use of the CICIoT2023 dataset, application of ML to IoT security, and inclusion of CKC-stage classification. No reviewed study satisfies all three simultaneously: studies using CICIoT2023 achieve strong detection performance but stop at attack-type classification (Section 2.1), while the one study to attempt CKC classification does so on non-IoT data (Section 2.3). This gap motivated the present study. The present study addresses it only in part: its stage labels are derived from the attack categories, so it evaluates a stage-level output on IoT traffic but does not infer stages from traffic evidence beyond the category (Section 3.3.3). Evaluation practice is a further concern. Arp et al. [10] identify data snooping, spurious correlations, the base-rate fallacy and lab-only evaluation among the pitfalls common in machine-learning-based security research; Kapoor and Narayanan [11] document data leakage as a widespread cause of

irreproducible results across machine-learning-based science; and Engelen et al. [12] show that feature-extraction and labelling errors in the CICIDS2017 intrusion detection dataset affect reported results. To our knowledge, none of the CICIoT2023 studies reviewed above reports a duplicate audit or a grouped evaluation, which the present study provides.

Table 1. Coverage of dataset, ML-based IoT security and CKC classification across the reviewed literature. [a] Partial: the stage is derived deterministically from the predicted attack category and evaluated as an output layer; no stage is inferred from traffic evidence beyond the category, and no temporal lifecycle is inferred (Section 3.3.3).

Study	CICIoT2023 Dataset	ML-Based IoT Security	CKC Stage Output
Gheni and Al-Yaseen (2024) [14]	✓	✓	×
Hajjouz and Avksentieva (2024) [15]	✓	✓	×
Kumar, Rastogi and Ranga (2024) [16]	✓	✓	×
Premalatha and Ramanujam (2025) [17]	✓	✓	×
Neto et al. (2023) [13]	✓	✓	×
Balega et al. (2024) [18]	×	✓	×
Mahdi (2024) [19]	×	✓	×
M et al. (2024) [20]	×	✓	×
Rashid et al. (2020) [21]	×	✓	×
Soliman et al. (2023) [22]	×	✓	×
El-Sofany et al. (2024) [23]	×	✓	×
Ahmed et al. (2021) [24]	×	×	✓
This study (2026)	✓	✓	(✓) [a]

3. Methodology

This study followed a two-phase quantitative design. Phase 1 develops and compares seven supervised classifiers for intrusion detection in IoT network traffic under a leakage-controlled evaluation protocol, and replicates the comparison on an independently processed release of the source capture. Phase 2 evaluates a Cyber Kill Chain stage-level output layer built on the Phase 1 feature set and LightGBM configuration. Explainability analysis with SHAP is applied to the leakage-controlled model. Both phases use the CICIoT2023 six-class release as their common foundation. All experiments were implemented in Python 3.13 using pandas, NumPy, scikit-learn, LightGBM, XGBoost, SHAP, statsmodels, DuckDB, matplotlib and seaborn. Figure 1 summarises the overall research methodology.

3.1. Dataset Description

The CICIoT2023 six-class dataset [13], developed by the Canadian Institute for Cybersecurity, was used for both phases of this study. It is a reduced, class-balanced derivative of the original CICIoT2023 dataset (45,807,617 instances across eight classes), retaining 1,800,000 network traffic instances distributed approximately evenly across six classes (benign traffic and five attack categories: DDoS, DoS, reconnaissance, Mirai and Web-based attacks). The dataset was selected in preference to conventional intrusion-detection benchmarks such as NSL-KDD [25] and UNSW-NB15 [26] because it was collected specifically from IoT traffic and captures attack types characteristic of consumer and SOHO device compromise. It was obtained as a single CSV file (df_FL_CICIoT2023_6_classes.csv, ≈1.2 GB uncompressed) from a public Zenodo repository [27], with file integrity confirmed against the repository's published MD5 checksum prior to use. The SHA-256 checksum of the file analysed is given in the code repository (Data Availability Statement). The independently processed release of the full capture used for external evaluation (Section 4.4) contains 46,776,697 records; record counts differ between separately processed versions of the capture.

Each instance is described by 17 fields: 16 numerical traffic features (flow duration, header length, protocol type, packet/byte rates, TCP flag counts, packet-size statistics, inter-arrival time, magnitude and variance) and one categorical label, sub_label, encoding benign traffic and five attack categories as shown in Table 2.

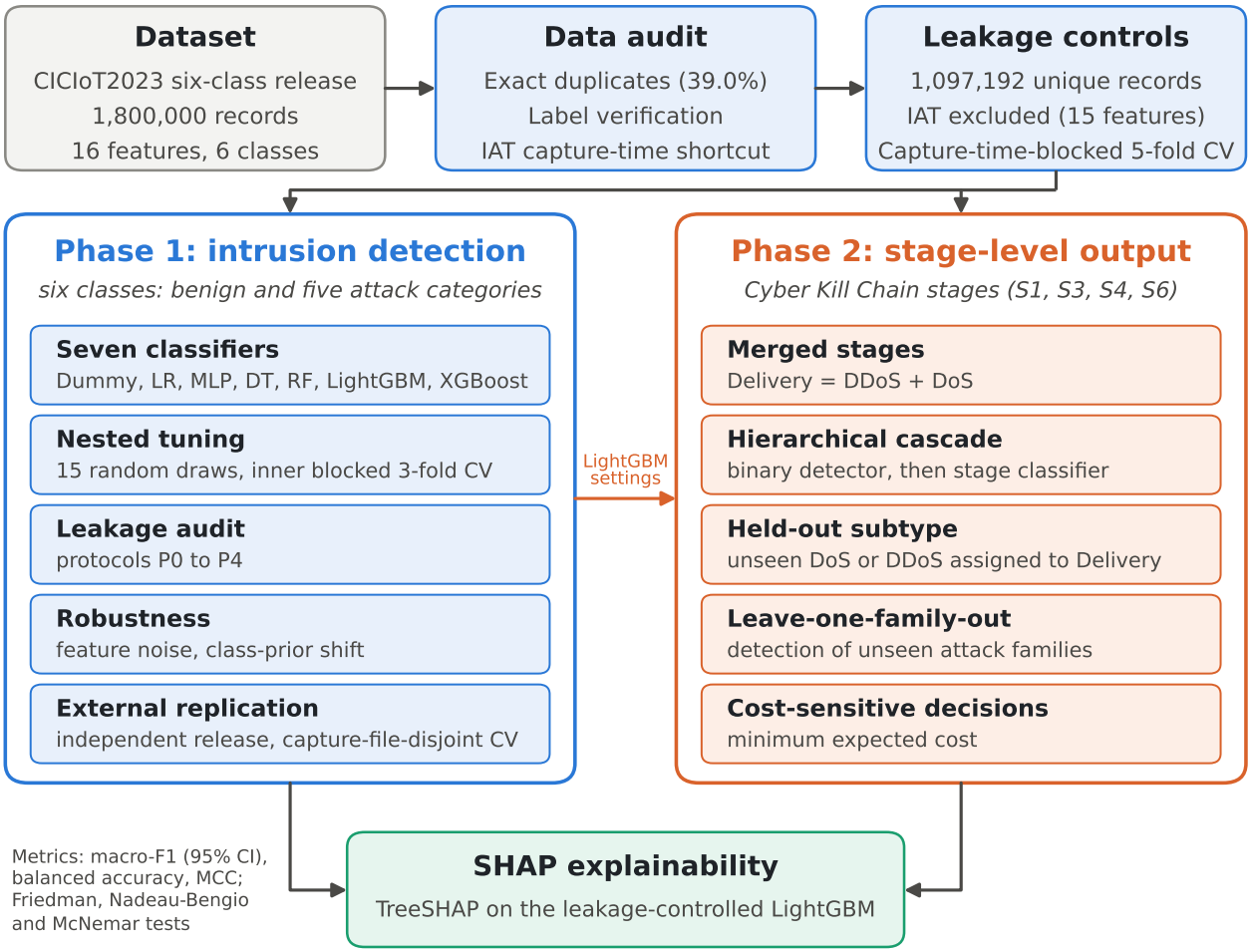


Figure 1. Overview of the research methodology: data audit and leakage controls, Phase 1 leakage-controlled intrusion detection with external replication, Phase 2 stage-level output, and SHAP explainability.

Table 2. Attack categories represented by the sub_label field (inferred mapping; see Section 3.1.2).

Sub_Label	Category	Constituent Attack Types
0	Benign	None
1	DDoS	ACK fragmentation, UDP flood, SlowLoris, ICMP flood, RSTFIN flood, PSHACK flood, HTTP flood, UDP fragmentation, TCP flood, SYN flood, SynonymousIP flood
2	DoS	TCP flood, HTTP flood, SYN flood, UDP flood
3	Mirai	GREIP flood, GREETH flood, UDPPlain
4	Reconnaissance	Ping sweep, OS scan, vulnerability scan, port scan, host discovery
5	Web attack	SQL injection, command injection, backdoor malware, upload attack, XSS, browser hijacking

The released file is near-uniform across classes (each class ≈ 16.67% of instances). The audit below shows, however, that this balance was produced by duplicating records, so all leakage-controlled experiments use the deduplicated, naturally imbalanced data. Table 3 describes each of the 16 numerical features in turn.

Table 3. Description of the numerical features in the dataset.

Feature	Description
`flow_duration`	Duration of the network flow (s)
`Header_Length`	Total length of packet headers (bytes)
Protocol Type	Encoded network protocol (e.g., TCP = 6, UDP = 17, GRE = 47)
`Rate`, `Srate`	Overall and source packet transmission rate (packets/s); the two columns are identical in this release
`syn_count`, `urg_count`, `rst_count`	Counts of TCP SYN, URG, and RST flags in the flow
Tot sum, Min, Max, AVG, Tot size	Packet-size statistics for the flow
`IAT`	Inter-arrival time between packets
Magnitude, Variance	Derived packet-size magnitude and variance measures

3.1.1. Data Audit

Before modelling, the file was audited for duplicate records. Of the 1,800,000 records, 702,808 (39.0%) are exact duplicates of another record, leaving 1,097,192 unique records. No identical feature vector carries conflicting labels. Duplication is concentrated in the classes that are rare in the original CICIoT2023 capture. This indicates that the release was balanced by oversampling with replacement (Table 4, Figure 2). After rounding each feature to three significant figures, a further 328,858 unique records (30.0%) collapse onto another record, so near-duplicates are also common. All leakage-controlled experiments therefore use the 1,097,192 unique records, which restores the classes' natural imbalance.

Table 4. Exact-duplicate audit of the CICIoT2023 six-class release.

Class (Sub_Label)	Records	Unique	Exact Duplicates	Duplicate Share
0 Benign	300,000	119,416	180,584	60.2%
1 DDoS	300,000	300,000	0	0.0%
2 DoS	300,000	300,000	0	0.0%
3 Mirai	300,000	286,086	13,914	4.6%
4 Reconnaissance	300,000	38,442	261,558	87.2%
5 Web attack	300,000	53,248	246,752	82.3%
Total	1,800,000	1,097,192	702,808	39.0%

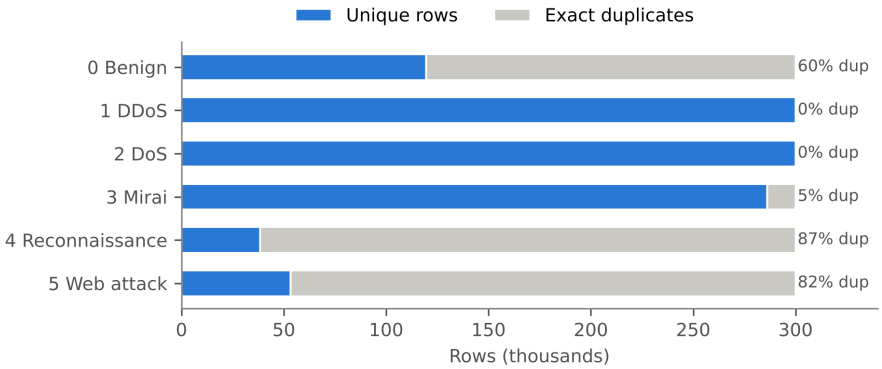


Figure 2. Unique and duplicated records per class (inferred class names; see Section 3.1.2).

3.1.2. Label Mapping: Basis and Verification

The six-class file carries only an integer label and is distributed without documentation of its encoding; neither the repository from which the file was obtained [27], nor the dataset paper [13], nor the original release defines these integers. No authoritative source therefore maps the integers to categories, and the mapping used here is inferred on two independent grounds. The first is the encoding convention. The order 0 Benign, 1 DDoS, 2 DoS, 3 Mirai, 4 Recon, 5 Web is the alphabetical order of the CICIoT2023 category names, which is the order assigned by standard label encoders such as scikit-learn's LabelEncoder [28]; the map used in the original manuscript does not follow this order. The second, set out below, is the agreement between the traffic of each class and the attacks that the dataset authors document for each category [13]. The label-to-category map was then verified against traffic characteristics that are diagnostic of each category in CICIoT2023 (Table 5). Class 3 is the only class containing GRE traffic (66% of records GRE-dominated; the remainder largely UDP), and its packets have a near-constant size (median 573 bytes, inter-quartile range 554–587). This matches the Mirai GREIP, GREETH and UDPPlain floods, which use a 512-byte default payload plus encapsulation headers; two of these three documented Mirai attacks are GRE-based, in line with the two-thirds share of GRE-dominated records. Class 4 shows the probing signature of reconnaissance: the smallest packets of any non-flood class (median 98 bytes; 62% of records with minimum packet size $\leq$ 60 bytes) and SYN and RST flags present in 89% and 94% of records, consistent with port and host scanning, the ping sweep, OS scan, vulnerability scan, port scan and host discovery that make up the documented Reconnaissance category. Class 4 contains no GRE traffic, whereas every Mirai attack in CICIoT2023 is a flood; under the original map, the Mirai class would consist of small, SYN- and RST-bearing probes and the Reconnaissance class of GRE floods, which contradicts the dataset documentation. Class 5 carries application payloads (median 364 bytes; 53% of records with a packet $\geq$ 1000 bytes) similar to benign web traffic, consistent with web attacks. The inferred map is therefore 0 Benign, 1 DDoS, 2 DoS, 3 Mirai, 4 Reconnaissance and 5 Web attack. This corrects the earlier version of this manuscript, which had interchanged classes 3 and 4. All results in this paper use the corrected map.

Table 5. Label verification and corrected label map (unique records). Classes 3 and 4 are the corrected rows. Constituent attacks as documented by the dataset authors [13]; the integer-to-category assignment is inferred (Section 3.1.2).

#	Original Manuscript	Inferred Category	CKC Stage	GRE-Dom.	UDP	Median pkt Size (B)	Min pkt $\leq$ 60 B	SYN > 0	RST > 0
0	Benign	Benign	Baseline	0%	0%	245	31%	66%	99%
1	DDoS	DDoS	S3 Delivery	0%	12%	54	96%	28%	23%
2	DoS	DoS	S3 Delivery	0%	25%	54	89%	35%	25%
3	Reconnais-sance	Mirai	S6 C2	66%	20%	573	0%	14%	24%
4	Mirai	Reconnais-sance	S1 Recon	0%	0%	98	62%	89%	94%
5	Web attack	Web attack	S4 Exploitation	0%	13%	364	30%	34%	85%

3.2. Preprocessing and Evaluation Protocol

A common preprocessing pipeline was applied in both phases: a dropna() check for missing values (none were found) and verification of dataset integrity and structure (via info() and describe()). The original analysis used an 80/20 stratified train–test split (*stratify* = *y*, *random_state* = 42; 1,440,000 training and 360,000 test instances). This protocol is retained only as the reference point P0 in the leakage audit (Section 4.1).

3.2.1. Shortcut Feature

In this release, IAT does not behave as a packet inter-arrival time. Its values cluster in narrow, class-specific bands of about 8.3×10^7 and 1.66×10^8 that track capture time. For example, the 1st–99th percentile ranges of DoS and DDoS records are nearly disjoint ($8.29–8.30 \times 10^7$ vs. $8.30–8.35 \times 10^7$). A depth-12 decision tree trained on IAT alone reaches 87.4% accuracy under the original random split (96.3% on a random split of the deduplicated records) and separates DDoS, DoS and Mirai (class 3) almost perfectly. IAT is therefore excluded as a model input in all leakage-controlled experiments. It is used only to construct the capture-time-blocked folds described next.

3.2.2. Capture-Time-Blocked Cross-Validation

The six-class file exposes no device, session or timestamp identifiers, so a true session-disjoint split cannot be built. As the closest available proxy, the records of each class were ranked by IAT and cut into five contiguous blocks of equal size. Fold k tests on block k of every class and trains on the remaining four, so the training and test records of each class come from disjoint capture-time ranges while all classes stay present in every fold. Every model is evaluated under five protocols of increasing strictness: (P0) the original random 80/20 split on all records; (P1) a random split on deduplicated records; (P2) P1 without IAT; (P3) capture-time-blocked CV with IAT; and (P4) capture-time-blocked CV without IAT. P4 is the primary protocol, and all main results are reported under it.

3.2.3. Exact-Copy Check

For P0, every test record was checked for an identical feature vector in the training partition. Accuracy is reported separately for copied and genuinely unseen test records.

3.3. Phase 1 and Phase 2 Design

3.3.1. Phase 1 Design

Phase 1 addresses detection: each flow is classified as benign or as one of the five attack categories, with `sub_label` as the six-class target. All leakage-controlled Phase 1 experiments use the 1,097,192 unique records and 15 of the 16 numerical features (IAT excluded), evaluated with capture-time-blocked five-fold cross-validation (protocol P4) and the classifiers and nested tuning described in Section 3.5. The original protocol (all records, all 16 features, random 80/20 split) is retained only as a reference point in the leakage audit (Section 4.1). Phase 1 is also replicated on an independently processed release of the full CICIoT2023 capture, using capture-file-disjoint folds (Section 4.4).

3.3.2. Phase 2 Design

A six-label CKC target that keeps DDoS and DoS as separate rows is a one-to-one relabelling of `sub_label`. Trained with the same algorithm, seed and data, it produces predictions identical to Phase 1 (verified: 100% agreement), so it is not reported as a separate task. Phase 2 instead evaluates what a stage-level output adds beyond an attack-category output, using five analyses under protocol P4: (i) a merged-stage model with a single Delivery stage (DDoS + DoS), compared with Phase 1 predictions collapsed to stages; (ii) a hierarchical cascade, in which a binary benign/malicious detector is followed by a stage classifier trained only on attack records, with detector errors propagated end to end; (iii) held-out-subtype generalisation, in which the stage model is trained without one Delivery subtype (DoS or DDoS) and evaluated on that unseen subtype, which is the only setting in which a stage label carries information the attack label cannot; (iv) leave-one-attack-family-out detection, measuring how often a binary detector flags an attack family absent from training; and (v) cost-sensitive stage assignment, which replaces the arg-max

decision with the minimum-expected-cost decision under a cost matrix (missed attack = 10, false alarm = 1, wrong stage = 2).

Benign traffic was retained as a non-attack baseline class rather than mapped onto the kill chain. Table 6 summarises this mapping and its rationale.

Phases 1 and 2 are implemented as two independent classifiers trained on the same flow-level feature set, rather than as a sequential pipeline in which Phase 2 consumes the output of Phase 1 (the hierarchical cascade, in which a stage classifier consumes the output of a binary detector, is the one exception). Phase 1 addresses the detection question (whether a flow is malicious and of which category), while Phase 2 addresses the contextualisation question (to which stage of the kill chain the observed behaviour corresponds). Because `Kill_Chain_Stage` is a deterministic re-expression of `sub_label`, Phase 2 does not infer an attack's position in a temporal lifecycle from traffic evidence alone; its purpose is to evaluate what defender-facing, stage-level output adds beyond category-level output (Section 4.5). Figure 1 shows the two phases as parallel branches that share the data audit and leakage controls; Phase 2 reuses the LightGBM configuration selected in Phase 1.

Table 6. Mapping of CICIoT2023 attack categories onto Cyber Kill Chain stages. Note: DDoS and DoS are both mapped to Stage 3 – Delivery and are merged into a single Delivery stage in the Phase 2 stage-level models. The Mirai records are flood traffic emitted by bots under C2 control rather than the C2 channel itself, so Actions on Objectives is a defensible alternative reading.

Attack Category	CKC Stage	Rationale
Benign	Not mapped (baseline)	Legitimate traffic; retained only to differentiate normal from malicious behaviour
DDoS	Stage 3–Delivery	Flood traffic is delivered directly to the victim to exhaust resources and deny service
DoS	Stage 3–Delivery	As DDoS, but launched from a single source rather than a distributed botnet [29]
Reconnaissance	Stage 1–Reconnaissance	Probing of ports, services, and hosts to gather information prior to exploitation
Mirai	Stage 6–Command & Control	Compromised devices connect to a C2 infrastructure for persistent remote control
Web attack	Stage 4–Exploitation	Exploits web-service vulnerabilities (e.g., SQL injection, XSS) to gain unauthorised access

The mapping in Table 6 is a conservative, category-level abstraction rather than an event-level attribution. Reconnaissance activity (port, service and host scanning) corresponds directly to Stage 1 of the Lockheed Martin kill chain [8], and web-application attacks such as SQL injection and cross-site scripting constitute exploitation of a vulnerable service and are therefore mapped to Stage 4. Mirai traffic in CICIoT2023 is generated by bot nodes operating under botnet control, which is the defining behaviour of Stage 6, Command and Control [3]. In CICIoT2023 these records are the floods launched by such bots (GREIP, GREETH and UDPPlain), so the label denotes traffic from a C2-controlled bot rather than the C2 channel itself. Volumetric DoS and DDoS traffic is the case requiring most care: the payload is the flood itself, which is conveyed to the target over the network, and delivery is therefore taken, following convention in kill-chain-based DDoS and IoT analyses, as the stage at which the adversary transmits its effect to the victim. DoS and DDoS may equally be characterised as Actions on Objectives where denial of service is the adversary's end goal, consistent with the MITRE ATT&CK framework, which places network and endpoint denial of service under its Impact tactic [30], and a botnet flood is simultaneously evidence of prior command and control. The mapping is accordingly a defender-facing labelling convention rather than an assertion of unique stage membership; extended models such as the Unified Kill Chain [31] likewise recognise that real attacks need not progress linearly through discrete phases, and the alternative readings for each category are recorded in the note to Table 6.

Because the mapping is deterministic and category-level, the stage labels inherit any ambiguity present in the original attack taxonomy. This constraint is recorded as a limitation

in Section 5.6, and session- and capture-level labelling with event-level ground truth is identified in Section 6 as the necessary next step towards genuine lifecycle inference.

3.3.3. Scope of the Cyber Kill Chain Interpretation

Because each attack category maps to exactly one stage (Table 6), a stage label is a deterministic function of the category label. Any category classifier followed by this lookup produces the same stage output, and stage-level accuracy is at least as high as category-level accuracy by construction, because confusions between categories that share a stage are not counted as errors. The stage-level output is therefore not an independent inference of kill-chain stage, and this study does not claim a new stage classifier. Its contribution is narrower: it measures what changes when results are reported at stage granularity, namely whether a confusion that the features cannot resolve disappears when both categories share a stage, whether an unseen subtype is still assigned to the correct stage, and how stage-aware costs change the balance between missed attacks and false alarms.

Three points define what the stage-level analysis does and does not claim.

What a Stage Label Is

The stage is a property of the attack category, assigned by the fixed mapping in Table 6. Every flow of a category receives that category's stage, and no stage is inferred from evidence beyond the category. Throughout the paper, including the "Cyber Kill Chain stage mapping" of its title, "stage classification" and "stage-level output" refer to this category-level assignment.

What Phase 2 Measures

Phase 2 measures how accurately individual flows are assigned to the stage of their category, and what reporting the stage adds for a defender. Because the mapping is many-to-one (DDoS and DoS share the Delivery stage), stage-level accuracy can exceed category-level accuracy, which is the effect reported for the merged Delivery stage. Only the held-out-subtype analysis tests whether a model recognises stage-relevant behaviour beyond the category labels it was trained on, because there the stage is the only correct output the model can give.

What Is Not Claimed

Phase 2 does not infer an attack's position in a temporal lifecycle, the order of its stages, or which flows belong to the same campaign. Flows are classified independently, with no linkage by session, host or time, and CICIoT2023 records each attack as a separate capture rather than as a progression. Four of the seven stages are represented, and Command and Control only indirectly (Table 7). Lifecycle inference would require session-labelled captures in which the same adversary progresses through successive stages.

Table 7. Coverage of the seven Cyber Kill Chain stages by the CICIoT2023 six-class labels.

Stage	Category	Coverage and Observability in Flow Features
1 Reconnaissance	Reconnaissance	Represented. Probing packets are observable; the intent behind a probe is not.
2 Weaponisation	None	Not represented. Takes place on the attacker's side and generates no traffic to the victim network.
3 Delivery	DDoS, DoS	Represented. The flood is itself the delivered effect (alternative reading: Actions on Objectives).
4 Exploitation	Web attack	Represented. Payload-bearing requests resemble benign web traffic.
5 Installation	None	No separate label. Backdoor-malware and upload attacks are grouped within the Web-attack category.
6 Command and Control	Mirai	Represented indirectly. The records are floods sent by bots under C2 control, not the C2 channel itself.
7 Actions on Objectives	None	No separate label. Denial of service could also be read as this stage.

3.4. Exploratory Data Analysis

3.4.1. Dataset Structure and Feature Distributions

The info() and describe() outputs confirmed 1,800,000 complete records with 16 float64 features and one int64 label, with no missing values, requiring no categorical encoding. Table 8 reports the describe() statistics for all 16 features. Several features exhibited strongly right-skewed distributions with long tails, notably Rate, Srate, and flow_duration, consistent with a small number of high-intensity flooding events (DoS/DDoS) amid a much larger volume of low-rate benign and reconnaissance traffic.

Table 8. Descriptive statistics of the 16 numerical features of the CICIoT2023 six-class release (all 1,800,000 records).

Feature	Mean	Std. Dev.	Min	Median	Max
`flow_duration` (s)	77.8	1069	0	1.20	68,431
`Header_Length`	530,158	1.25×10^6	0	9568	9.89×10^6
Protocol Type	12.9	12.5	0	7.10	47.0
`Rate`	5664	72,154	0	33.9	8.39×10^6
`Srate`	5664	72,154	0	33.9	8.39×10^6
`syn_count`	0.44	0.68	0	0	12.2
`urg_count`	73.9	265.4	0	0	4192
`rst_count`	386.0	1005	0	0.30	9613
Tot sum	4007	6025	50.0	1243	116,053
Min	194.9	298.6	42.0	60.0	5858
Max	776.2	1313	42.0	230.0	30,329
AVG	382.0	534.0	42.0	114.1	8280
Tot size	382.3	537.0	42.0	109.5	13,098
`IAT`	8.35×10^7	5.79×10^7	0	8.31×10^7	1.67×10^8
Magnitue	22.7	15.4	9.17	15.0	126.5
Variance	0.45	0.44	0	0.24	1.00

3.4.2. Feature Correlation Analysis

A clustered correlation heatmap of all 16 numerical features (Figure 3) was used to assess redundancy prior to modelling. Two clusters and a weakly correlated remainder emerged.

A packet-size cluster (Magnitue, AVG, Tot sum, Tot size, Max) showed strong mutual correlation ($r = 0.70$–0.96), reflecting shared measurement of packet-size volume; Min is moderately correlated with it ($r = 0.29$–0.69). A header/flag cluster (Header_Length, rst_count, urg_count) showed moderate-to-strong correlation ($r = 0.63$–0.87), associated with header structure and control-flag behaviour. The remaining features (flow_duration, Rate, Srate, IAT, Protocol Type, syn_count, Variance) are weakly correlated with one another and with the two clusters ($|r| \leq 0.45$), with one exception: Rate and Srate are identical in this release ($r = 1.00$), so one of them is redundant. Because tree-based ensembles such as Random Forest, Decision Tree, and LightGBM are largely insensitive to multicollinearity, all features were retained (including both Rate and Srate, to keep the feature set identical to the original analysis) rather than removed or combined; IAT was later excluded for a different reason, namely its behaviour as a capture-time stamp (Section 3.2).

3.4.3. Class Distribution

The value_counts(normalize = True) method confirmed a near-uniform class distribution, with each of the six classes representing approximately 16.67% of the full dataset (≈300,000 instances per class), preserved as 60,000 instances per class in the stratified test split (Figure 2). This balance, however, was produced by duplication (Section 3.1.1); after deduplication the classes are imbalanced (Table 4), and macro-averaged metrics are therefore used throughout. A balanced test set also characterises performance under a prior that

real SOHO traffic does not exhibit: in deployment, benign flows dominate and attack traffic is rare and unevenly distributed across categories. Accuracy measured under a balanced test prior therefore overstates operational performance, and precision in particular would fall as the benign prior rises even with recall held constant. Macro-averaged metrics are consequently reported alongside accuracy, together with a prior-shift sensitivity analysis computed analytically for attack prevalence of 10%, 1% and 0.1%, so that the effect of realistic imbalance on precision and false-positive volume is visible. Extension to naturally imbalanced traffic is identified as future work in Section 6.

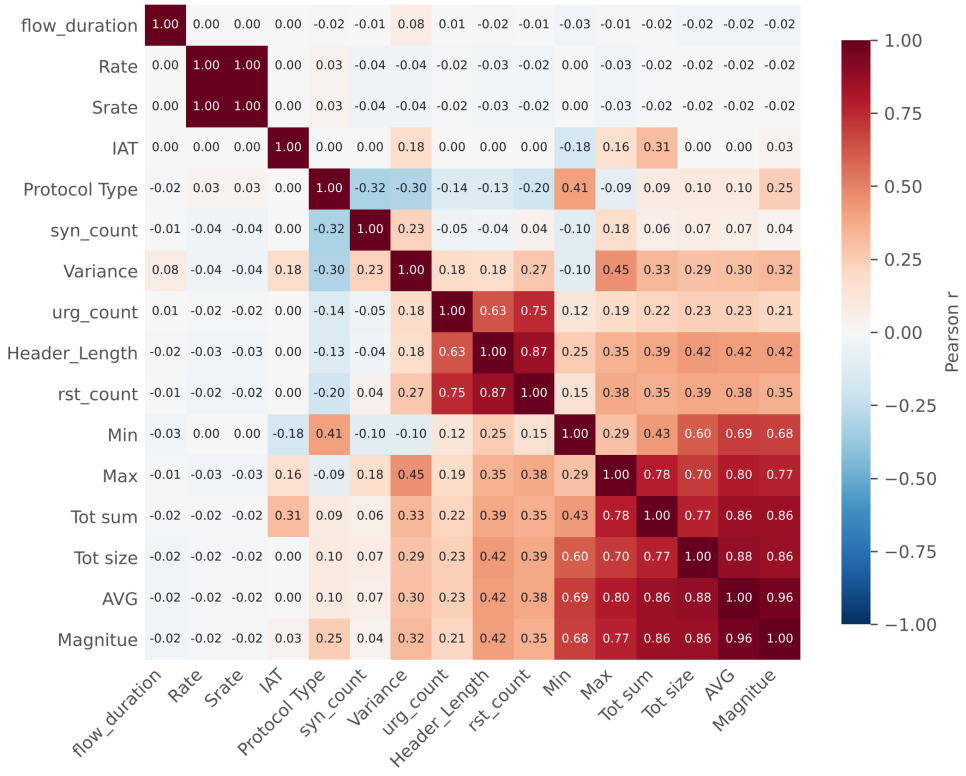


Figure 3. Pearson correlation of the 16 numerical features (all 1,800,000 records), ordered by average-linkage clustering on $1 - |r|$.

3.5. Model Implementation

Seven classifiers were compared: a stratified dummy classifier (chance baseline); multinomial logistic regression with a normal-quantile transform and balanced class weights; a multilayer perceptron (128–64 ReLU units, Adam, early stopping, quantile-transformed inputs); Decision Tree; Random Forest [32]; LightGBM [33]; and XGBoost [34] (histogram method). The four tree-based models were tuned by nested randomised search. Within each outer training fold, 15 configurations were sampled from the spaces in Table 9. Each configuration was scored by macro-F1 on an inner three-fold capture-time-blocked split of a stratified subsample (up to 8000 records per class) of that training fold only. The best configuration was then refitted on the full outer training fold. Test folds were never used for tuning. Phase 2 uses LightGBM with the hyperparameters selected for each fold in Phase 1. LightGBM is carried forward because it is statistically indistinguishable from the best Phase 1 model (Section 4.2) while being more than ten times smaller than Random Forest (Section 4.2).

Table 9. Hyperparameter search spaces (15 random draws per outer fold).

Model	Search Space (15 Random Draws per Outer Fold)
Decision Tree	`max_depth` ∈ {None, 10, 15, 20, 30}; `min_samples_leaf` ∈ {1, 2, 5, 10, 20, 50}; `criterion` ∈ {gini, entropy}; `class_weight` ∈ {None, balanced}
Random Forest	`n_estimators` ∈ {100, 200}; `max_depth` ∈ {None, 15, 25}; `min_samples_leaf` ∈ {1, 2, 5, 10}; `max_features` ∈ {sqrt, 0.5}; `class_weight` ∈ {None, balanced_subsample}
LightGBM	`n_estimators` ∈ {200, 400, 800}; `learning_rate` ∈ {0.03, 0.05, 0.1}; `num_leaves` ∈ {31, 63, 127, 255}; `min_child_samples` ∈ {10, 30, 100}; `subsample` ∈ {0.7, 1.0}; `colsample_bytree` ∈ {0.6, 0.8, 1.0}; `reg_lambda` ∈ {0, 1, 5}; `class_weight` ∈ {None, balanced}
XGBoost	`n_estimators` ∈ {200, 400, 800}; `learning_rate` ∈ {0.05, 0.1, 0.2}; `max_depth` ∈ {6, 8, 10, 12}; `min_child_weight` ∈ {1, 5}; `subsample` ∈ {0.7, 1.0}; `colsample_bytree` ∈ {0.6, 0.8, 1.0}; `reg_lambda` ∈ {1, 5}

3.6. Evaluation Metrics

Model performance was assessed on the held-out test folds of each protocol using accuracy, precision, recall, and F1-score, computed per class via scikit-learn's classification_report, together with confusion matrices for error analysis. For a given class, true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN) were defined in the conventional one-vs-rest sense. The metrics were computed as:

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}$$

$$Precision = \frac{TP}{TP + FP}$$

$$Recall = \frac{TP}{TP + FN}$$

$$F1 = \frac{2 \times Precision \times Recall}{Precision + Recall}$$

These follow standard definitions: recall ([35]), precision ([36]), and F1-score ([37]). To interpret model decisions, SHAP (SHapley Additive exPlanations) [38] values and gain-based importance were computed for the leakage-controlled LightGBM model (Section 4.6), quantifying each feature's contribution to individual predictions.

Because the deduplicated data are imbalanced, the primary metric is macro-averaged F1, reported as the mean ± standard deviation over the five outer folds with a 95% confidence interval (t-distribution, 4 d.f.). Balanced accuracy, the Matthews correlation coefficient (MCC) [39] and per-class F1 are also reported. Models are compared with a Friedman test across folds [40], followed by Nadeau–Bengio corrected resampled t-tests [41] between the best model and each alternative; the correction accounts for overlap between training sets across folds. McNemar tests [42] on the pooled out-of-fold predictions are reported for completeness. With about 1.1 million paired predictions, McNemar detects even negligible differences, so the fold-level corrected test is used for inference. Robustness to measurement noise is assessed by multiplying each continuous test feature by $\exp(\varepsilon)$, $\varepsilon \sim \mathcal{N}(0, \sigma^2)$, $\sigma \in \{0.05, 0.1, 0.2, 0.5\}$. Sensitivity to class prior, which determines alert precision through the base-rate effect [10], is assessed analytically from the out-of-fold true-positive rate (TPR) and benign false-positive rate (FPR) of the derived binary attack decision:

$$Precision = \frac{TPR \cdot \pi}{TPR \cdot \pi + FPR \cdot (1 - \pi)}$$

for attack prevalence $\pi \in \{0.5, 0.1, 0.01, 0.001\}$. Training time, inference time per 1000 flows and serialised model size were measured on a 2-core, 8 GB virtual machine.

4. Analysis

4.1. Phase 1: Data Leakage Audit

Table 10 and Figure 4 show how much of the originally reported performance is attributable to leakage. Under the original protocol (P0), 48.6% of test records have an exact copy in the training partition, including 99.9% and 99.2% of the Reconnaissance and Web attack test records. Random Forest and Decision Tree classify every copied record correctly (100.0%). On genuinely unseen Reconnaissance and Web attack records, Random Forest recall falls to 84.1% and 82.8%. LightGBM, whose leaf-wise trees are constrained in depth, cannot memorise copies in the same way and scores 91.0% on copied records. This, rather than class balance, explains its lower P0 accuracy. Removing duplicates lowers Random Forest macro-F1 from 0.999 to 0.954 and nearly closes the gap to LightGBM (0.946). Excluding IAT lowers it further, to 0.842: without the capture-time feature, DDoS and DoS become largely indistinguishable (F1 $\approx$ 0.83/0.83). Under capture-time-blocked CV without IAT (P4), the three default-configured models reach 0.699–0.729, with fold-to-fold standard deviations of about 0.08. Keeping IAT under blocked CV (P3) does not help: IAT values in a held-out capture-time range lie outside the range seen in training, and Reconnaissance F1 collapses to 0.14–0.45.

Table 10. Macro-F1 of the original default-configured models under progressively stricter protocols (blocked CV: mean $\pm$ SD over five folds).

Protocol	Random Forest	Decision Tree	LightGBM
P0 Original: random 80/20, all records, all features	0.999	0.998	0.954
P1 Deduplicated, random 80/20	0.954	0.936	0.946
P2 Deduplicated, random 80/20, no IAT	0.842	0.804	0.839
P3 Deduplicated, capture-time-blocked 5-fold, with IAT	0.777 $\pm$ 0.130	0.690 $\pm$ 0.116	0.758 $\pm$ 0.082
P4 Deduplicated, capture-time-blocked 5-fold, no IAT	0.725 $\pm$ 0.079	0.699 $\pm$ 0.066	0.729 $\pm$ 0.086

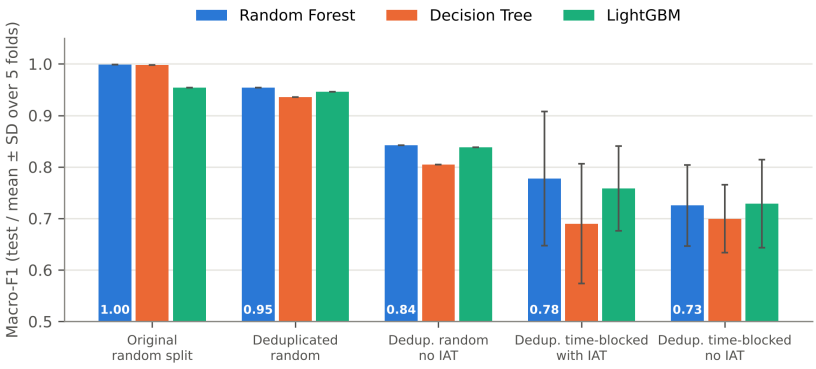


Figure 4. Macro-F1 under progressively stricter evaluation protocols. Each control removes a source of optimism: exact duplicates, then the IAT capture-time shortcut, then overlap in capture-time range between training and test data.

4.2. Phase 1: Leakage-Controlled Intrusion Detection

Table 11 and Figure 5 report the seven classifiers under protocol P4. All learned models clearly outperform the chance baseline (macro-F1 0.167), and logistic regression (0.519) confirms that the classes are not linearly separable. XGBoost achieves the highest mean macro-F1 (0.733 $\pm$ 0.079), followed by LightGBM (0.731) and Random Forest (0.729). The Friedman test (Table 12) indicates that the models differ overall ($\chi^2 = 20.89$, $p < 0.001$),

but the corrected resampled t-tests find no significant difference between XGBoost and LightGBM ($\Delta = 0.002$, $p = 0.67$) or Random Forest ($\Delta = 0.005$, $p = 0.70$), whereas Decision Tree ($p = 0.028$) and logistic regression ($p = 0.027$) are significantly worse. Nested tuning improved LightGBM only marginally over its default configuration (0.731 vs. 0.729), indicating that the remaining error comes from the data rather than from model configuration. Per-class results (Figure 6) show three regimes. Mirai is separated almost perfectly (F1 0.99), owing to its fixed-size GRE/UDP flood packets. DDoS and DoS are confused with each other in about a third of cases in each direction (F1 0.63/0.64), which indicates that, in the window-averaged features of this release, little information distinguishes distributed from single-source floods once IAT is removed. A similar confusion occurs in the independently processed release (Section 4.4), so it is not specific to the feature construction of the six-class release. Benign, Reconnaissance and Web attack traffic are confused with one another (F1 0.86, 0.57, 0.70). Reconnaissance is the hardest category: a single probe window is a short, low-rate TCP exchange that resembles a benign connection attempt. The large fold-to-fold variation (SD $\approx$ 0.08) shows that performance depends on which capture-time range is held out.

Table 11. Leakage-controlled results (protocol P4: deduplicated, IAT excluded, capture-time-blocked 5-fold CV, nested tuning). c0 = Benign, c1 = DDoS, c2 = DoS, c3 = Mirai, c4 = Reconnaissance, c5 = Web attack (verified map, Section 3.1.2).

Model	Macro-F1 (Mean ± SD)	95% CI	Bal. acc.	MCC	c0	c1	c2	c3	c4	c5
XGBoost (tuned)	0.733 ± 0.079	±0.098	0.731	0.694	0.86	0.63	0.64	0.99	0.57	0.70
LightGBM (tuned)	0.731 ± 0.085	±0.106	0.729	0.693	0.86	0.63	0.64	0.99	0.57	0.70
Random Forest (tuned)	0.729 ± 0.082	±0.102	0.733	0.691	0.85	0.62	0.65	0.99	0.60	0.66
Decision Tree (tuned)	0.701 ± 0.089	±0.110	0.706	0.672	0.82	0.62	0.63	0.99	0.52	0.63
MLP (128–64)	0.695 ± 0.065	±0.081	0.694	0.652	0.84	0.57	0.61	0.99	0.53	0.64
Logistic regression	0.519 ± 0.037	±0.045	0.572	0.456	0.63	0.44	0.33	0.91	0.42	0.39
Stratified dummy	0.167 ± 0.000	±0.000	0.167	0.001	0.11	0.27	0.27	0.26	0.03	0.05

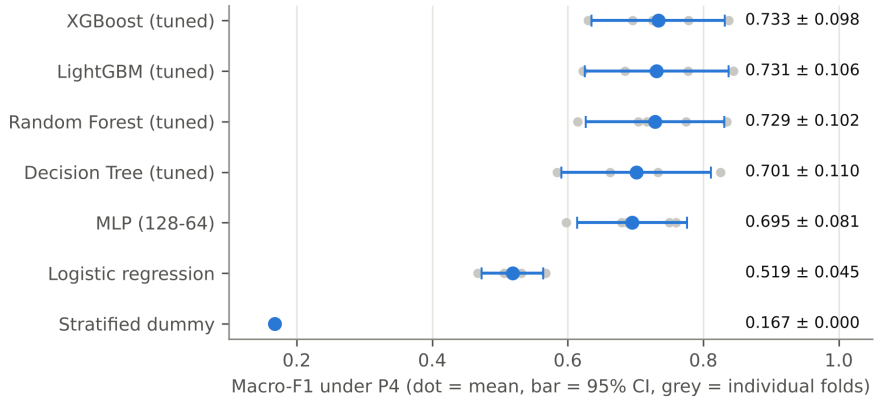


Figure 5. Macro-F1 per model under protocol P4: mean, 95% confidence interval and the five individual folds.

Table 12. Pairwise comparisons with the best model. Friedman test across all six learned models: $\chi^2 = 20.89$, $p < 0.001$.

Comparison	Δ Macro-F1	Corrected t	p (Nadeau–Bengio)	McNemar χ^2	p (McNemar)
XGBoost vs. Logistic regression	0.215	3.42	0.027	141,040	<0.001
XGBoost vs. MLP (128–64)	0.038	2.25	0.088	14,622	<0.001
XGBoost vs. Decision Tree	0.032	3.37	0.028	3,657	<0.001
XGBoost vs. Random Forest	0.005	0.42	0.699	148	<0.001
XGBoost vs. LightGBM	0.002	0.45	0.675	29	<0.001

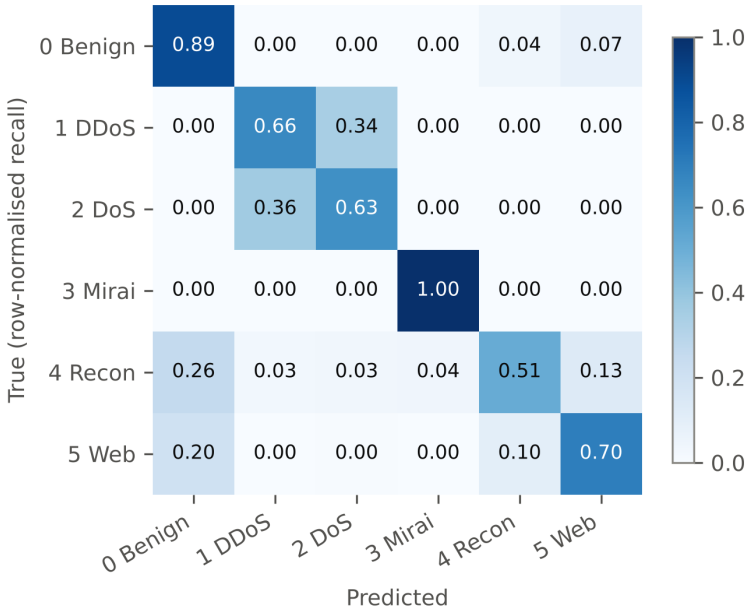


Figure 6. Row-normalised out-of-fold confusion matrix for XGBoost under protocol P4.

Computational Cost

Mean training time per fold was 16 s (Decision Tree), 126 s (Random Forest), 170 s (LightGBM) and 223 s (XGBoost) on about 880,000 records. Serialised model sizes were 2.98 MB, 733 MB, 40 MB and 54 MB respectively, and inference took 0.12, 7.68, 222.05 and 68.81 ms per 1000 flows. The three ensembles are statistically equivalent in accuracy but differ by more than an order of magnitude in memory footprint, which matters more for constrained SOHO hardware than the accuracy differences do.

4.3. Phase 1: Robustness to Noise and Class Prior

4.3.1. Feature Noise (Figure 7)

Under multiplicative noise with $\sigma = 0.2$, macro-F1 falls by 0.036 for Random Forest, 0.084 for LightGBM, 0.078 for XGBoost and 0.152 for Decision Tree. At $\sigma = 0.5$, Random Forest retains 0.621, against 0.526–0.547 for the boosted models. Bagging therefore gives the most robust model, and a single tree the least robust.

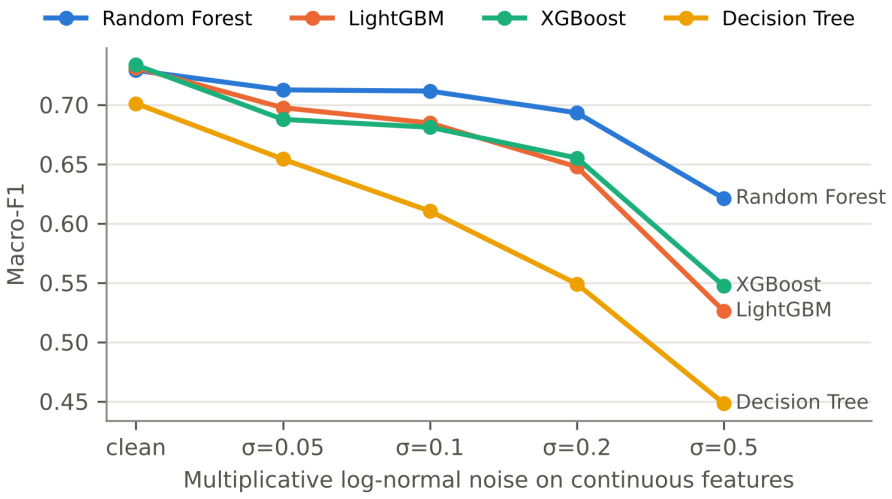


Figure 7. Macro-F1 under multiplicative log-normal noise on the 14 continuous features (mean over five folds).

4.3.2. Class Prior (Figure 8, Table 13)

For the binary attack decision, the best model detects 97.9% of attack flows but flags 10.9% of benign flows. Under the balanced test prior, its precision is 90.0%. At 10%, 1% and 0.1% attack prevalence, precision falls to 49.9%, 8.3% and 0.9%, the last corresponding to about 109,223 false alerts per million flows. By contrast, the leaky original protocol implied a benign false-positive rate of 0.51% for Random Forest. A detector at the measured operating point would overwhelm a SOHO user with alerts. Deployment would require threshold calibration to a target false-alarm rate, and a two-stage design in which only high-confidence alerts are escalated.

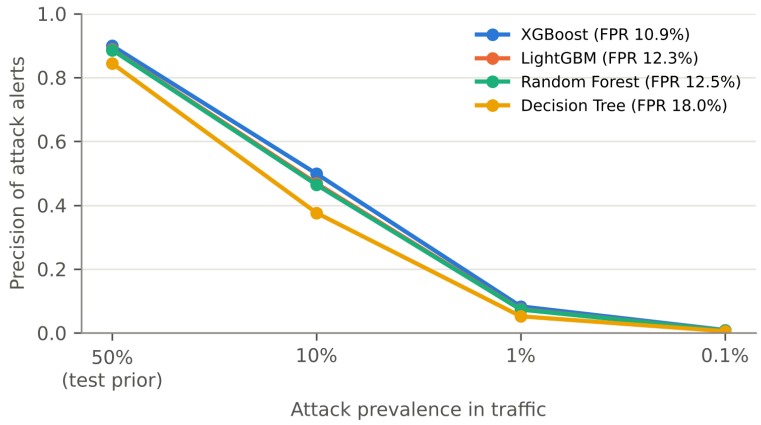


Figure 8. Precision of the binary attack decision as attack prevalence falls (analytical, from out-of-fold TPR and FPR).

Table 13. Prior-shift sensitivity of the derived binary attack decision (protocol P4).

Model	TPR	Benign FPR	Precision @50%	@10%	@1%	@0.1%	False Alerts/1M @1%
XGBoost	97.9%	10.9%	90.0%	49.9%	8.3%	0.89%	108,239
LightGBM	98.0%	12.3%	88.8%	46.9%	7.4%	0.79%	122,167
Random Forest	97.8%	12.5%	88.6%	46.4%	7.3%	0.77%	124,165
Decision Tree	97.8%	18.0%	84.4%	37.6%	5.2%	0.54%	178,458
MLP (128–64)	97.4%	11.9%	89.1%	47.5%	7.6%	0.81%	118,220
Logistic regression	97.8%	44.7%	68.6%	19.5%	2.2%	0.22%	442,887

4.4. Phase 1: External Evaluation on an Independently Processed Release

4.4.1. Data and Protocol

To test whether the Phase 1 conclusions hold beyond the six-class file, the pipeline was repeated on a separately processed version of the full CICIoT2023 capture: 46,776,697 records extracted per capture file with 39 numeric features (38 traffic features and an indicator for rates recorded as infinite) (integrity verified against the SHA-256 checksum recorded in its cleaning report). This release uses different feature definitions: un-averaged packet windows, per-packet rather than cumulative header length, and a genuine inter-arrival time in seconds. Models trained on the six-class file therefore cannot be applied to it directly, and the same model configurations were retrained on it instead (tuned settings from Section 4.2, no further tuning). Spoofing and brute-force records were excluded to match the six classes. A deterministic, file-balanced sample was drawn with about 150,000 records per category, spread evenly across capture files (747,267 records, including all 24,829 Web-attack records). Unlike the six-class release, this sample contains no exact duplicates. Each record carries its source capture file, which allows a genuinely grouped evaluation. Within each category, capture files were assigned at random to four folds, so that no test file

contributes training records. Benign traffic spans 4 files, DDoS 176, DoS 38 and Mirai 76. Each Reconnaissance (5) and Web-attack (6) subtype occupies a single file, so file-disjoint testing for these categories also means that the test subtype was unseen in training. IAT and time-to-live (TTL) were excluded, leaving 37 features; the ablation below shows that neither acts as a shortcut in this release. Columns that record a record's position within its capture file were not used, because they identify the capture and the position in it rather than traffic behaviour.

4.4.2. Results

Under a random record-level split, the three models reach macro-F1 0.774–0.803 (Table 14). Under capture-file-disjoint evaluation, macro-F1 falls to 0.720 ± 0.015 for LightGBM and XGBoost and to 0.697 ± 0.013 for Random Forest. The record-level split therefore overstates performance by about 0.08 even though this release contains no exact duplicates, because records from the same capture file are highly similar. Per class, the pattern of Section 4.2 recurs (Figure 9). Mirai is separated almost perfectly (F1 1.00). DDoS and DoS are confused with each other (F1 0.79 and 0.82): pooled over folds, 25.8% of DDoS flows are predicted as DoS and 12.8% of DoS flows as DDoS, and the confusion is also present under the random split (24.4% and 7.4%). Benign, Reconnaissance and Web attack traffic are confused with one another (F1 0.80, 0.59 and 0.33): 30.4% of reconnaissance flows are classified as benign, and of the flows from an unseen web-attack subtype, 47.9% are classified as reconnaissance and 27.1% as benign. Removing IAT, TTL or both changes LightGBM macro-F1 by less than 0.005 (Table 15). The external evaluation thus confirms three conclusions of Sections 4.1 and 4.2. First, record-level splits overstate performance even without duplicates. Second, the flow features separate DDoS from DoS only partially in both releases. Third, reconnaissance and application-level web attacks, whose flows resemble benign traffic, are the weak point. The three ensembles again perform similarly, with Random Forest slightly behind LightGBM and XGBoost. Absolute values are not directly comparable with Table 11, because the feature definitions and the grouping unit differ.

Table 14. External evaluation on the independently processed CICIoT2023 release: random record-level split vs. capture-file-disjoint 4-fold evaluation (per-class F1 = mean over file-disjoint folds; IAT and TTL excluded).

Model	Random Split Macro-F1	File-Disjoint Macro-F1 (Mean ± SD)	F1 Benign	F1 DDoS	F1 DoS	F1 Mirai	F1 Recon.	F1 Web Attack
LightGBM	0.803	0.720 ± 0.015	0.80	0.79	0.82	1.00	0.59	0.33
XGBoost	0.802	0.720 ± 0.015	0.80	0.79	0.82	1.00	0.59	0.33
Random Forest	0.774	0.697 ± 0.013	0.78	0.77	0.77	1.00	0.57	0.29

Table 15. Shortcut-feature ablation on the external release (LightGBM macro-F1).

LightGBM Feature Set	Random Split	File-Disjoint (Mean ± SD)
All 39 features	0.809	0.724 ± 0.016
Without IAT	0.809	0.724 ± 0.018
Without TTL	0.804	0.721 ± 0.015
Without IAT and TTL	0.803	0.720 ± 0.015

4.5. Phase 2: Stage-Level Output

4.5.1. Relabelling Equivalence and Merged Delivery Stage

As expected, a stage model with one Delivery stage behaves like Phase 1 with DDoS–DoS confusions forgiven. Its stage-level macro-F1 is 0.824, against 0.823 for Phase 1 predictions collapsed to stages, and the two agree on 99.2% of flows (Table 16). The merged

Delivery stage reaches F1 1.00, compared with 0.63/0.64 for the separate DDoS and DoS classes in the six-class model. This is the first concrete benefit of the stage abstraction: it removes a distinction that the flow features of either release cannot reliably support and that is operationally irrelevant at the stage level (both are volumetric delivery of a denial-of-service effect). The same benefit appears in the independently processed release: with DDoS and DoS merged, the Delivery stage reaches F1 0.998–0.999 in every capture-file-disjoint fold (against 0.79/0.82 for the separate classes, Table 14), and the stage-level macro-F1 of the LightGBM predictions is 0.743, against 0.720 at category level.

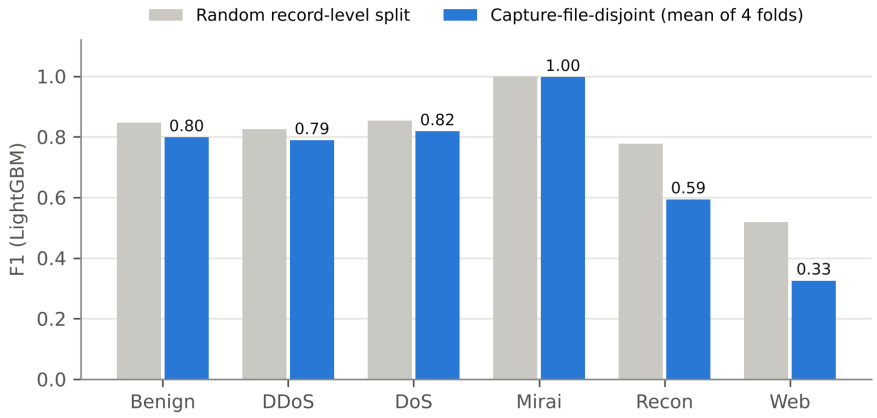


Figure 9. Per-class F1 of LightGBM (IAT and TTL excluded) on the independently processed release: random record-level split vs. capture-file-disjoint evaluation.

4.5.2. Hierarchical Cascade

The binary detector flags 97.9% of attack flows with a benign false-positive rate of 9.6%. Passing its positives to an attack-only stage classifier gives an end-to-end stage macro-F1 of 0.831 (balanced accuracy 0.821), slightly above the flat stage model (0.824). The accuracy gain is marginal; the main benefit is that the cascade separates the detection decision (where the false-alarm threshold is set) from stage attribution, which suits deployment.

4.5.3. Held-Out Delivery Subtype

When DoS is entirely absent from training, 90.9% of DoS flows are still assigned to the Delivery stage (7.7% to Exploitation, 0.0% to benign). When DDoS is held out, 97.9% of DDoS flows are assigned to Delivery. A six-class attack model cannot output a label it has never seen, so this is the clearest evidence that stage-level output generalises beyond the training taxonomy within a stage.

4.5.4. Unseen Attack Families (Table 17)

In leave-one-family-out detection, the binary detector flags 100% of DDoS, 100% of DoS, 100% of Mirai, 63% of Reconnaissance, 31% of Web attack flows when that family is absent from training, with benign false-positive rates of 4–10%. The volumetric families (DDoS, DoS, Mirai) generalise from one to another. Reconnaissance and Web attack flows, which resemble benign traffic, are largely missed when unseen.

4.5.5. Cost-Sensitive Stage Decisions

Replacing the arg-max decision with the minimum-expected-cost decision lowers the mean cost per flow from 0.228 to 0.137. Missed attacks fall from 20,004 to 4877, at the price of more false alarms (14,736 → 54,940). Stage-level probabilities thus support explicit, defender-controlled trade-offs that a single accuracy figure hides.

Table 16. Stage-level results under protocol P4 (out-of-fold, tuned LightGBM configuration per fold).

Approach	Stage Macro-F1	Bal. acc.	F1 Benign	F1 S3 Delivery	F1 S6 C2	F1 S1 Recon	F1 S4 Exploit.
Phase 1 (6 classes) collapsed to stages	0.823	0.816	0.86	1.00	0.99	0.56	0.71
Flat merged-stage model	0.824	0.816	0.86	1.00	0.99	0.56	0.71
Hierarchical cascade (detector → stage)	0.831	0.821	0.87	1.00	0.99	0.59	0.71
Cost-sensitive (min. expected cost)	0.760	0.789	0.68	1.00	0.99	0.52	0.61

Table 17. Leave-one-attack-family-out binary detection (mean over five capture-time-blocked folds; training capped at 100,000 records per class).

Held-Out Family (Class)	Recall on Unseen Family	Recall on Seen Families	Benign FPR
1 DDoS	100.0%	97.1%	10.3%
2 DoS	100.0%	97.1%	10.2%
3 Mirai	100.0%	97.1%	10.3%
4 Reconnaissance	63.2%	98.5%	7.6%
5 Web attack	31.3%	98.6%	4.4%

4.6. Explainability

SHAP values were recomputed for the tuned LightGBM model of capture-time-blocked fold 0 (IAT excluded), using exact TreeSHAP [38,43] (tree-path-dependent feature perturbation). The explained set was a stratified sample of 3000 held-out test records (500 per class, seed 1). Per-class SHAP matrices were aggregated as the mean absolute value over classes and records. The most influential features are Tot sum, Protocol Type, Magnitude, Variance, Min (Table 18, Figure 10): packet-size volume and protocol composition, rather than timing. SHAP and gain rankings differ, which is expected: gain is the total loss reduction from the splits on a feature during training, whereas SHAP measures each feature's contribution to individual predictions. In the original analysis, IAT, `Header_Length` and `flow_duration` ranked first to third. The prominence of IAT reflected the capture-time artefact, not a timing signature of the attacks.

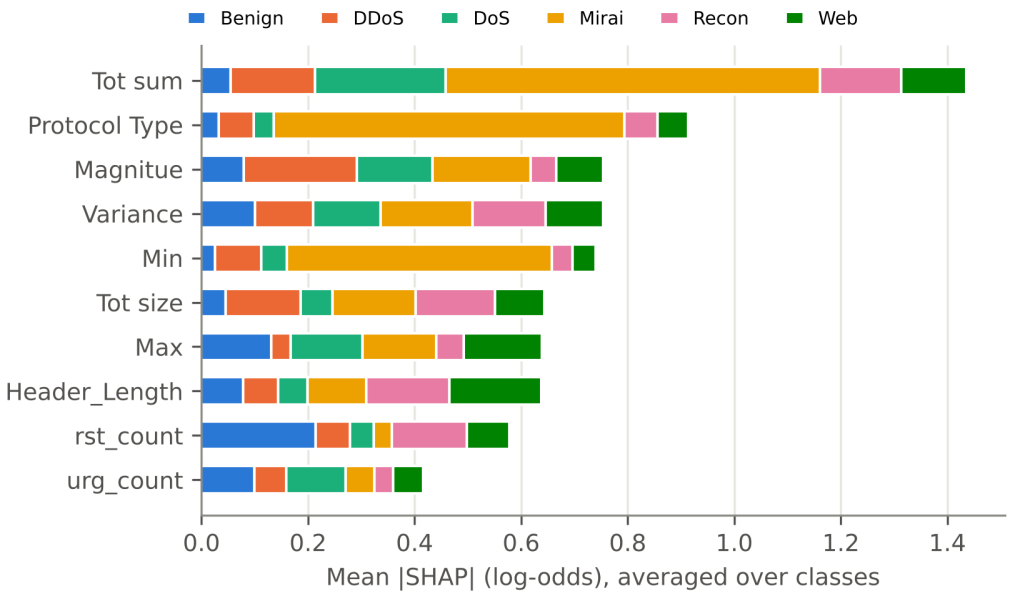


Figure 10. Mean absolute SHAP value per feature, split by class, for the leakage-controlled LightGBM model.

Table 18. Feature influence for the leakage-controlled LightGBM model.

Rank (SHAP)	Feature	Mean |SHAP|	Gain Rank
1	Tot sum	1.436	13
2	Protocol Type	0.913	9
3	Magnitude	0.754	6
4	Variance	0.754	5
5	Min	0.740	1
6	Tot size	0.643	7
7	Max	0.639	3
8	`Header_Length`	0.638	10
9	`rst_count`	0.578	2
10	`urg_count`	0.416	8

5. Discussion

5.1. Interpretation of Intrusion Detection Findings

The ranking reported under the conventional random split (Random Forest > Decision Tree > LightGBM) was an artefact of duplication. Fully grown trees memorise records that reappear in the test partition, while LightGBM's depth-constrained trees do not. Once duplicates were removed, the ranking reversed. Under the leakage-controlled protocol, the three ensembles are statistically indistinguishable (macro-F1 0.729–0.733). Among them, Random Forest is the most robust to feature noise, while LightGBM and XGBoost are more than ten times smaller.

Taken together, the results do not support a single preferred classifier on accuracy grounds. Random Forest is the most robust to feature noise, whereas LightGBM and XGBoost generalise slightly better across capture files (Section 4.4) and offer equivalent accuracy at a far smaller model size, which matters for constrained SOHO hardware.

5.2. Interpretation of Stage-Level Findings

With the corrected label map and leakage-controlled evaluation, the difficulty ordering is the reverse of the one originally reported. Mirai traffic (Stage 6) is separated almost perfectly (F1 0.99) because, in CICIoT2023, it consists of GRE and UDP floods with fixed-size packets, not of low-volume keep-alive messages. Reconnaissance (Stage 1) is the hardest category (F1 0.57), followed by Web exploitation (Stage 4, F1 0.70). Both are confused mainly with benign traffic. The data therefore do not support the claim that late kill-chain stages are harder to detect. What they show is that low-volume, protocol-conformant activity is hard to detect at any stage, which matters for defenders because reconnaissance is the stage at which early intervention is most valuable.

The stage-level analysis shows where a kill-chain abstraction adds value on flow data. It removes the DDoS/DoS distinction, which the flow features of either release cannot reliably resolve. It assigns an unseen Delivery subtype to the correct stage. It exposes cost trade-offs between missed attacks and false alarms. It does not infer an attack's position in a temporal lifecycle: the stage labels are a deterministic, category-level mapping, and the dataset contains no session-level progression against which such inference could be validated.

To our knowledge, no prior study has evaluated a kill-chain stage-level output on IoT traffic: the closest work [24] maps security alerts from advanced persistent threats (APTs), not network flows, to stages. The novelty is limited accordingly. It lies in evaluating how a category-derived stage output behaves on IoT flows, not in inferring stages: outside the held-out-subtype setting, the stage labels carry no information beyond the attack categories. The scope of the stage-level claims is set out in Section 3.3.3.

5.3. Explainability Insights

The integration of SHAP alongside gain-based feature importance served two purposes: clarifying which traffic characteristics drive the assignment of flows to attack categories, and therefore to the stages those categories map to, and surfacing which characteristics are most diagnostic of each category. The original prominence of IAT is a cautionary example for explainable intrusion detection. A feature can dominate SHAP and gain rankings because it encodes how and when the data were captured, not how attacks behave. Explanations computed on a leaky model can look plausible and still be wrong, an instance of the spurious correlations described by Arp et al. [10]. Explainability does not replace leakage-controlled validation. A further limitation is that SHAP and gain rankings can disagree (Table 18), which complicates their interpretation by non-specialist defenders, an issue also raised in the wider explainable-AI-for-cybersecurity literature [44].

5.4. Comparison with Related Literature

Most prior CICIoT2023 results in Figure 11 were obtained with random record-level splits, and many of the derived releases are balanced by oversampling. Those figures are therefore likely to share the optimism quantified in Section 4.1, a pattern consistent with the wider evidence on leakage in machine-learning-based research [11], and they are not like-for-like comparators for the leakage-controlled results reported here. Under the original protocol, our Phase 1 figures (99.86% accuracy for Random Forest) were in line with this literature: ref. [14] reported 97.26% binary and 96.26% multiclass accuracy using a multilayer perceptron, ref. [15] 99.4% for DoS/DDoS subtypes using CatBoost, and [16] near-100% accuracy for Decision Tree and Random Forest on grouped attack categories. Our audit shows that such figures can arise from record-level splitting of duplicated data: under leakage-controlled evaluation, the same models reach a macro-F1 of about 0.73 (accuracy 76.1% for XGBoost). Beyond CICIoT2023, comparable accuracies have been reported on other datasets, for example [18] with XGBoost on IoT-23/NSL-KDD (99.98%) and [22] with Bi-LSTM on ToN-IoT (99.99%).

Accuracy figures drawn from different datasets, task definitions, class structures and splitting protocols are not directly comparable. Figure 11 therefore provides contextual reference points rather than a like-for-like benchmark; only the CICIoT2023 six-class studies are treated as near-comparable, and even there differences in feature selection and split protocol apply.

For CKC-specific classification, the closest comparator remains [24], whose APT-focused Naive Bayes classifier achieved 91.1% accuracy. Our stage-level output reaches 95.3% accuracy under leakage-controlled evaluation (macro-F1 0.82; Table 16), but the two figures are not comparable: the data, the tasks and the label definitions differ, and our stage labels are assigned at category level (Section 3.3.3). The contribution here is therefore an evaluation of a stage-level output layer on IoT traffic, an area explicitly identified as under-researched by [9], rather than a higher stage accuracy.

5.5. Theoretical, Practical, and Policy Implications

Theoretically, this study extends IoT IDS research beyond binary or multiclass attack detection toward stage-aware output, responding to calls for kill-chain-informed detection frameworks [9], applies SHAP to a leakage-controlled IoT intrusion classifier whose categories carry stage labels, and shows that, once a capture-time artefact is removed, packet-size volume and protocol composition rather than timing drive the model's decisions.

Practically, at the measured operating point (benign false-positive rate $\approx$ 11%), flow-level classifiers trained on this release are not ready for unsupervised deployment on SOHO routers. Their value lies in triage and stage-aware prioritisation, combined with calibrated

thresholds and human or rule-based confirmation. Stage-aware classification enables more differentiated incident response, for example, escalating alerts differently depending on whether traffic reflects reconnaissance, a volumetric Delivery-stage flood or flood traffic from bots under command and control.

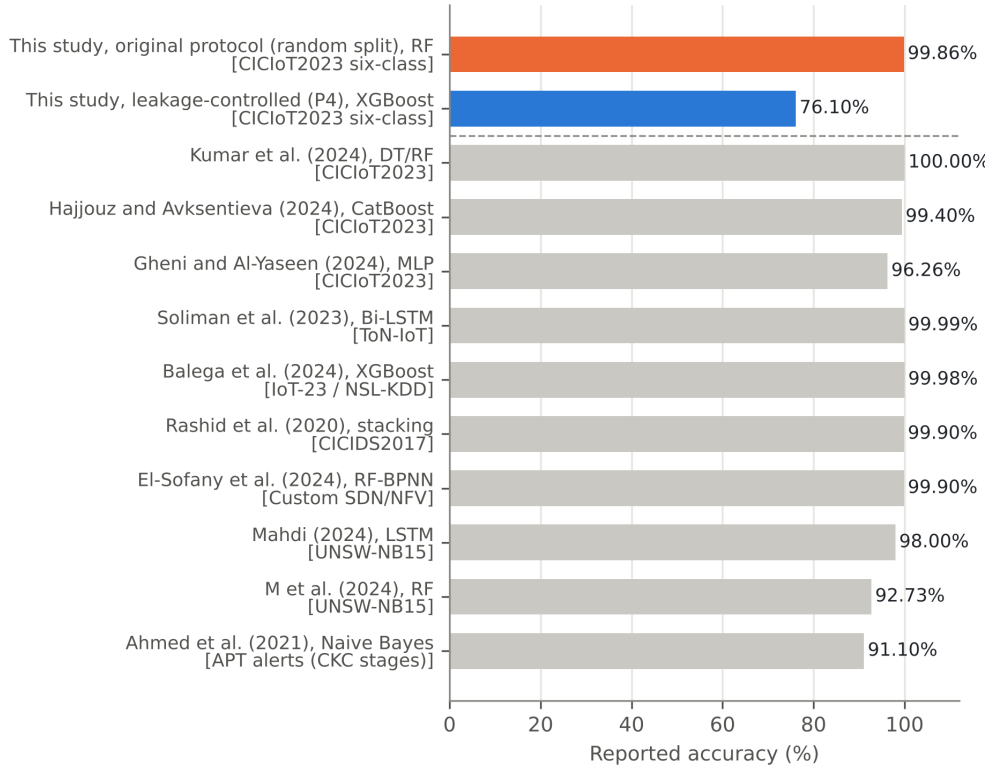


Figure 11. Reported accuracy in the reviewed literature [14–16,18–24] (**grey**) and in this study under the original random-split protocol (**orange**) and the leakage-controlled protocol P4 (**blue**). Datasets, tasks and splitting protocols differ, so the bars are contextual reference points rather than a like-for-like benchmark.

At a policy level, the findings support arguments for embedding baseline detection capability into consumer IoT devices and SOHO routers. This is consistent with warnings from the UK National Cyber Security Centre regarding the systematic exploitation of SOHO devices [4] and with existing consumer-IoT security guidance such as the UK Code of Practice for Consumer IoT Security [45]. Before such capability could be mandated in internet service provider or device-manufacturer security requirements, however, the low alert precision measured at realistic attack prevalence (Section 4.3) would need to be addressed through calibrated thresholds and confirmation mechanisms. The continued exploitation of SOHO devices in large-scale botnet campaigns [5] makes this worth pursuing.

5.6. Strengths and Limitations

We can summarise the strengths of our study as follows:

- An evaluation of a CKC stage-level output layer on IoT traffic, an area with very limited prior coverage in the CICIoT2023 literature, with the scope of the stage-level claims stated explicitly (Section 3.3.3); the stages are derived from the attack categories rather than inferred.
- A leakage-controlled evaluation (deduplication, shortcut-feature exclusion, capture-time-blocked and capture-file-disjoint validation) that quantifies how much reported performance is attributable to leakage.

- Integration of SHAP-based explainability, supporting transparency and trust in the classifier's decisions.
- Explicit framing around SOHO environments, a segment often overlooked relative to enterprise or industrial IoT security research.

Several limitations should be considered when interpreting these findings:

- CICIoT2023 was collected in a controlled testbed; real-world SOHO traffic may exhibit greater noise and device heterogeneity, limiting generalisability.
- Only four of the seven canonical CKC stages (reconnaissance, delivery, exploitation, command-and-control) were represented, owing to constraints in the underlying dataset labels, and command and control only indirectly (Table 7).
- Among neural models, only a shallow MLP was evaluated; recurrent and convolutional architectures (e.g., LSTM, CNN) were not, despite reported state-of-the-art results for such models in related IDS work.
- SHAP analysis, while informative, remains computationally intensive and non-trivial to interpret for non-specialist practitioners.
- Random Forest, the most noise-robust ensemble, has a much larger model than LightGBM (733 MB vs. 40 MB), which may constrain deployment on the lowest-power SOHO hardware.
- Capture-time blocks derived from IAT are a proxy for session-disjoint evaluation. They do not guarantee device- or session-level independence, so the reported figures may still be somewhat optimistic.
- The Reconnaissance and Web-attack classes retain only 38,442 and 53,248 unique records, so their estimates are the least stable (largest fold-to-fold variation).
- External evaluation uses an independently processed release of the same CICIoT2023 testbed, not a different network. Validation on traffic from other IoT testbeds remains future work.
- Reconnaissance and Web-attack subtypes each occupy a single capture file, so file-disjoint evaluation of these categories is also subtype-disjoint. Their external scores therefore measure generalisation to unseen subtypes, a stricter test than for the other categories.
- Training time, inference time and model size were measured on a 2-core, 8 GB virtual machine, not on SOHO router hardware; no deployment or throughput testing on routers or embedded devices was performed.
- The released dataset is artificially balanced and, after deduplication, imbalanced in a way that does not reflect operational SOHO traffic, which is dominated by benign flows; prior-shift effects were assessed analytically rather than on naturally imbalanced traffic.
- Kill chain stage labels are deterministically derived from the original attack categories, so Phase 2 demonstrates stage-level, defender-facing classification rather than inference of an attack's position in a temporal lifecycle; CICIoT2023 contains no session-level progression against which such inference could be validated.
- The category-to-stage mapping is a labelling convention and admits alternative readings, notably DoS and DDoS as Actions on Objectives, and Mirai flood traffic as evidence of both command and control and delivery; event-level ground truth would be required to resolve these.

6. Conclusions

This study developed and evaluated a two-phase, ML-based intrusion detection system for SOHO IoT traffic that adds a Cyber Kill Chain stage-level output to conventional attack classification, supported by explainable AI. On the class-balanced CICIoT2023 six-class release, near-perfect accuracy arises from duplicate records and a capture-time shortcut

feature. Under a leakage-controlled protocol with nested tuning, the best model reaches a macro-F1 of 0.733, the three tree ensembles are statistically indistinguishable, and benign false-positive rates of about 11% imply low alert precision at realistic attack prevalence. A kill-chain stage-level output adds measurable value: it absorbs the DDoS/DoS confusion present in both releases, generalises to an unseen Delivery subtype and supports cost-sensitive decisions. It does not, however, constitute temporal lifecycle inference. On an independently processed release of the full capture, capture-file-disjoint evaluation confirms both the optimism of record-level splits and the weakness on reconnaissance and unseen web-attack subtypes (macro-F1 0.720 vs. 0.803). SHAP analysis of the leakage-controlled model identifies packet-size volume and protocol composition, rather than timing, as the main drivers of its decisions. The study contributes academically by demonstrating, for IoT traffic specifically, that a stage-level output derived from attack categories can be evaluated, explained and used for cost-aware decisions when evaluated without leakage, partly addressing a gap between conventional binary/multiclass IDS research and kill-chain-informed defence frameworks. Practically, it shows how much of the accuracy reported for such detectors depends on the evaluation protocol, and that stage-level output can support differentiated, cost-aware responses, while the SHAP-based analysis offers defenders interpretable insight into which traffic characteristics drive model decisions. At a policy level, the findings are consistent with calls for embedding baseline, stage-aware detection capability into consumer IoT devices and SOHO routers. As no deployment, throughput or on-device resource testing on a physical router, gateway or embedded device was undertaken, SOHO environments constitute the motivating application context and the basis for dataset selection rather than a validated deployment target, and these practical and policy implications are conditions to be satisfied before deployment rather than demonstrated outcomes. These contributions should be read alongside the limitations discussed in Section 5.6, particularly the controlled-environment origin of the dataset, the restricted coverage of CKC stages and the absence of on-device evaluation. Future work should prioritise: (i) on-device evaluation on representative SOHO hardware, reporting training time, inference latency, memory footprint and model size; (ii) validating the approach on real-world SOHO traffic with realistic class imbalance (predominantly benign flows), rather than the approximately balanced distribution used here; (iii) extending CKC coverage to additional stages, through synthetic data generation or supplementary datasets; (iv) evaluating deep learning architectures (e.g., LSTM, GRU, CNN) for their capacity to resolve the reconnaissance and web-attack classification challenges identified here; (v) exploring federated or edge-based training to address data-privacy constraints in SOHO settings; (vi) combining signature-based and ML-based detection for improved robustness; (vii) simplifying SHAP-based outputs into forms that are directly actionable for non-specialist practitioners; (viii) collecting session-labelled IoT captures that allow genuine stage-progression modelling; (ix) calibrating detection thresholds to a target false-alarm budget; and (x) external testing on independently captured traffic from other IoT testbeds.

Author Contributions: Conceptualization, B.A.; methodology, B.A.; validation, E.H.M. and U.O.; investigation, U.O.; writing—original draft preparation, B.A., E.H.M. and U.O.; writing—review and editing, B.A., E.H.M. and U.O.; visualization, U.O.; supervision, B.A.; project administration, B.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: The original data presented in the study are openly available on the Canadian Institute for Cybersecurity (CIC) Website at https://www.unb.ca/cic/datasets/iotdataset-2023.html (accessed on 16 November 2026). The analysis code, the selected hyperparameters for each fold and all result files are openly available at https://github.com/Estabraq-makiyah/ML-Based-Cyber-Kill-Chain-Classification-for-IoT-Devices-in-SOHO-Environments (accessed on 16 November 2026). The repository maps every table and figure to the script that produces it and to its result file, gives SHA-256 checksums of the input files, and contains the row index needed to rebuild the external evaluation sample exactly. All experiments use random seed 0 for tuning and model fitting and seed 42 for the reproduction of the original protocol.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Statista. *Number of IoT Connected Devices Worldwide 2019–2030*; Statista Research Department: Hamburg, Germany, 2024.
2. Trend Micro. *The State of IoT Security*; Trend Micro Research: Irving, TX, USA, 2020.
3. Antonakakis, M.; April, T.; Bailey, M.; Bernhard, M.; Bursztein, E.; Cochran, J.; Durumeric, Z.; Halderman, J.A.; Invernizzi, L.; Kallitsis, M.; et al. Understanding the Mirai botnet. In Proceedings of the 26th USENIX Security Symposium, Vancouver, BC, Canada, 16–18 August 2017; pp. 1093–1110.
4. NCSC. *The NCSC and Partners Issue Alert About Evolving Techniques Used by China State-Sponsored Cyber Attackers*; National Cyber Security Centre: London, UK, 2024.
5. The Hacker News. *Over 1000 SOHO Devices Hacked in China-Linked LapDogs Cyber Espionage Campaign*; The Hacker News: New York, NY, USA, 2025.
6. Extech Cloud. *Cyberattack Disables 600,000 SOHO Routers in US*; Extech Cloud: Burgess Hill, UK, 2024.
7. Anderson, H. Artificial intelligence and its role in modern cybersecurity. *J. Inf. Secur.* **2018**, *12*, 45–59.
8. Hutchins, E.M.; Cloppert, M.J.; Amin, R.M. *Intelligence-Driven Computer Network Defense Informed by Analysis of Adversary Campaigns and Intrusion Kill Chains*; Technical Report; Lockheed Martin: Bethesda, MD, USA, 2011.
9. EC-Council. The Cyber Kill Chain: The Seven Steps of a Cyberattack. Cybersecurity Exchange. 2022. Available online: https://www.eccouncil.org/cybersecurity-exchange/threat-intelligence/cyber-kill-chain-seven-steps-cyberattack/ (accessed on 16 November 2026).
10. Arp, D.; Quiring, E.; Pendlebury, F.; Warnecke, A.; Pierazzi, F.; Wressnegger, C.; Cavallaro, L.; Rieck, K. Dos and Don'ts of Machine Learning in Computer Security. In Proceedings of the 31st USENIX Security Symposium (USENIX Security 22), Boston, MA, USA, 10–12 August 2022; pp. 3971–3988.
11. Kapoor, S.; Narayanan, A. Leakage and the Reproducibility Crisis in Machine-Learning-Based Science. *Patterns* **2023**, *4*, 100804. [CrossRef] [PubMed]
12. Engelen, G.; Rimmer, V.; Joosen, W. Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study. In Proceedings of the 2021 IEEE Security and Privacy Workshops (SPW), Virtual, 27 May 2021. [CrossRef]
13. Neto, E.C.P.; Dadkhah, S.; Ferreira, R.; Zohourian, A.; Lu, R.; Ghorbani, A.A. CICIoT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment. *Sensors* **2023**, *23*, 5941. [CrossRef] [PubMed]
14. Gheni, H.Q.; Al-Yaseen, W.L. Two-step data clustering for improved intrusion detection system using CICIoT2023 dataset. *e-Prime Adv. Electr. Eng. Electron. Energy* **2024**, *8*, 100673.
15. Hajjouz, A.; Avksentieva, E. Optimizing intrusion detection for DDoS, DoS, and Mirai attack subtypes using hierarchical feature selection and CatBoost on the CICIoT2023 dataset. *Data Metadata* **2024**, *3*, 577. [CrossRef]
16. Kumar, A.G.; Rastogi, A.; Ranga, V. Evaluation of different machine learning classifiers on new IoT dataset CICIoT2023. In Proceedings of the 2024 International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP), Rome, Italy, 17–19 July 2024; pp. 1–6.
17. Premalatha, D.V.; Ramanujam, S. Ensemble-based intrusion detection for IoT networks using the CICIoT2023 dataset. *J. Inf. Syst. Eng. Manag.* **2025**, *10*, 743–755. [CrossRef]
18. Balega, M.; Farag, W.; Wu, X.W.; Ezekiel, S.; Good, Z. Enhancing IoT security: Optimizing anomaly detection through machine learning. *Electronics* **2024**, *13*, 2148. [CrossRef]
19. Mahdi, M.A. Secure and efficient IoT networks: An AI and ML-based intrusion detection system. In Proceedings of the 2024 IEEE International Conference on Artificial Intelligence, Internet of Things (AIIoT), Seattle, WA, USA, 29–31 May 2024; pp. 1–6.
20. Saumya, Y.M.; Vinay, P.; Pinto, C.; Correia, N.E.; Miranda, M.C.; Dsouza, J.R. SmartDefend—IoT security using machine learning. In Proceedings of the 2024 IEEE International Conference on Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER), Karnataka, India, 18–19 October 2024; pp. 109–114.

21. Rashid, M.M.; Kamruzzaman, J.; Hassan, M.M.; Imam, T.; Gordon, S. Cyberattacks detection in IoT-based smart city applications using machine learning techniques. *Int. J. Environ. Res. Public Health* **2020**, *17*, 9347. [CrossRef] [PubMed]
22. Soliman, S.; Oudah, W.; Aljuhani, A. Deep learning-based intrusion detection approach for securing industrial Internet of Things. *Alex. Eng. J.* **2023**, *81*, 371–383. [CrossRef]
23. El-Sofany, H.; El-Seoud, S.A.; Karam, O.H.; Bouallegue, B. Using machine learning algorithms to enhance IoT system security. *Sci. Rep.* **2024**, *14*, 12077. [CrossRef] [PubMed]
24. Ahmed, A.; Mahmood, A.; Hu, J. A machine learning framework for mapping alerts to cyber kill chain stages for APT detection. *J. Inf. Secur. Appl.* **2021**, *60*, 102860.
25. Available online: https://www.kaggle.com/datasets/hassan06/nslkdd (accessed on 16 November 2026).
26. Available online: https://www.kaggle.com/datasets/mrwellsdavid/unsw-nb15 (accessed on 16 November 2026).
27. STIC Laboratory . Preprocessed Datasets. Zenodo, 2024. Preprocessed CICIoT2023 and Edge-IIoTset Datasets for Binary and Multiclass Classification; Version v1, Published 26 May 2024. Available online: https://zenodo.org/records/11315294 (accessed on 16 November 2026).
28. Pedregosa, F.; Varoquaux, G.; Gramfort, A.; Michel, V.; Thirion, B.; Grisel, O.; Blondel, M.; Prettenhofer, P.; Weiss, R.; Dubourg, V.; et al. Scikit-learn: Machine Learning in Python. *J. Mach. Learn. Res.* **2011**, *12*, 2825–2830.
29. Mirkovic, J.; Reiher, P. A taxonomy of DDoS attack and defense mechanisms. *ACM SIGCOMM Comput. Commun. Rev.* **2004**, *34*, 39–53. [CrossRef]
30. MITRE Corporation. Impact (TA0040). MITRE ATT&CK Enterprise Matrix. 2019. Available online: https://attack.mitre.org/tactics/TA0040/ (accessed on 16 November 2026).
31. Pols, P. The Unified Kill Chain: Designing a Unified Kill Chain for Analyzing, Comparing and Defending against Cyber Attacks. Master's Thesis, Cyber Security Academy, The Hague, The Netherlands, 2017.
32. Breiman, L. Random Forests. *Mach. Learn.* **2001**, *45*, 5–32. [CrossRef]
33. Ke, G.; Meng, Q.; Finley, T.; Wang, T.; Chen, W.; Ma, W.; Ye, Q.; Liu, T.Y. LightGBM: A highly efficient gradient boosting decision tree. In Proceedings of the Advances in Neural Information Processing Systems (NeurIPS), Long Beach, CA, USA, 4–9 December 2017; Volume 30.
34. Chen, T.; Guestrin, C. XGBoost: A Scalable Tree Boosting System. In Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, San Francisco, CA, USA, 13–17 August 2016; pp. 785–794. [CrossRef]
35. Powers, D.M.W. Evaluation: From precision, recall and F-measure to ROC, informedness, markedness and correlation. *J. Mach. Learn. Technol.* **2011**, *2*, 37–63.
36. Han, J.; Kamber, M.; Pei, J. *Data Mining: Concepts and Techniques*, 3rd ed.; Morgan Kaufmann: Waltham, MA, USA, 2012.
37. Sammut, C.; Webb, G.I. (Eds.) *Encyclopedia of Machine Learning*; Springer: New York, NY, USA, 2010.
38. Lundberg, S.M.; Lee, S.I. A Unified Approach to Interpreting Model Predictions. In Proceedings of the Advances in Neural Information Processing Systems 30 (NeurIPS 2017), Long Beach, CA, USA, 4–9 December 2017; pp. 4765–4774.
39. Chicco, D.; Jurman, G. The Advantages of the Matthews Correlation Coefficient (MCC) over F1 Score and Accuracy in Binary Classification Evaluation. *BMC Genom.* **2020**, *21*, 6. [CrossRef] [PubMed]
40. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* **2006**, *7*, 1–30.
41. Nadeau, C.; Bengio, Y. Inference for the Generalization Error. *Mach. Learn.* **2003**, *52*, 239–281. [CrossRef]
42. McNemar, Q. Note on the Sampling Error of the Difference between Correlated Proportions or Percentages. *Psychometrika* **1947**, *12*, 153–157. [CrossRef] [PubMed]
43. Lundberg, S.M.; Erion, G.; Chen, H.; DeGrave, A.; Prutkin, J.M.; Nair, B.; Katz, R.; Himmelfarb, J.; Bansal, N.; Lee, S.I. From Local Explanations to Global Understanding with Explainable AI for Trees. *Nat. Mach. Intell.* **2020**, *2*, 56–67. [CrossRef] [PubMed]
44. Holzinger, A.; Saranti, A.; Molnar, C.; Biecek, P.; Samek, W. Explainable AI in cybersecurity: State-of-the-art and future directions. *Inf. Fusion* **2022**, *81*, 84–102.
45. DCMS. *Code of Practice for Consumer IoT Security*; Technical Report; UK Department for Digital, Culture, Media and Sport: London, UK, 2021.

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.