

Class-Aware Knowledge Distillation for Detecting Rare Attacks with Tiny IoT Intrusion Detection Models

Suhaila Abuowaida^{1*}

¹Department of Data Science and Artificial Intelligence, Faculty of Prince Al-Hussein, Bin Abdallah II for Information Technology, Al Al-Bayt University, Mafraq, 25113, Jordan.

Corresponding Author: Suhaila Abuowaida. Email: suhila@aabu.edu.jo

Received: June 03, 2026 Accepted: August 30, 2026

Abstract: Knowledge distillation (KD) is a common way to shrink an intrusion detection model until it fits on an Internet-of-Things (IoT) edge device. Accuracy is usually the only number reported after distillation, and on heavily imbalanced traffic it is dominated by flooding attacks. We show that this can hide a large loss on rare attack classes. On CIC-IoT2023, Edge-IIoTset and ToN-IoT, a 5–9K-parameter student distilled from a transformer teacher with standard KD loses 11.8, 26.4 and 65.1 points of minority-class recall relative to its teacher. On ToN-IoT, recall on ransomware falls from 69.2% to 0% while overall accuracy stays above 95%. Splitting this gap shows that the capacity gap between teacher and student is about 11 points on every dataset, while the loss caused by distillation itself, measured against the same student trained without a teacher, is 0.3, 14.9 and 54.3 points for KD at the setting we tested. We trace the effect to the teacher’s soft labels, which inherit the class prior, and propose Minority-Aware KD (MA-KD): a distillation term reweighted per class and computed against prior-debiased teacher logits, which we relate formally to Balanced Knowledge Distillation. MA-KD raises minority recall over standard KD by 10.8, 16.6 and 73.5 points and macro-F1 by 2.0, 11.1 and 15.6 points, with 95% intervals that exclude zero on every dataset. An ablation attributes the gain to class reweighting and prior debiasing; a per-class temperature schedule we also tested has no measurable effect. We also report a negative result: a student trained alone with class-weighted cross-entropy is at least as good as MA-KD, with higher macro-F1 on all three datasets and higher minority recall on Edge-IIoTset, so at this student size the teacher repairs the damage done by distillation but adds little beyond a well-configured imbalance-aware loss. The student occupies 21 KB, runs in 14 μ s per flow on one cloud CPU thread, and is 41–103 times faster than the teacher. A cross-dataset study on a 13-feature harmonized schema fails for every model, which we report as a limitation of current IoT benchmarks.

Keywords: Intrusion Detection; Internet of Things; Knowledge Distillation; Class Imbalance; Long-Tailed Learning; Lightweight Models; Edge Deployment

1. Introduction

Intrusion detection for the Internet of Things (IoT) has two demands that pull against each other. Deep models trained on flow records reach high detection rates, but gateways, cameras and industrial controllers have kilobytes of memory and little spare compute. Knowledge distillation (KD) [1,2] is a natural response: train a large teacher offline, then fit a small student to its softened predictions. A growing number of IoT intrusion detection systems (IDS) follow this recipe [3–5], and most report accuracy or weighted F1 after compression.

IoT traffic is extremely imbalanced. In CIC-IoT2023 [6] distributed denial-of-service floods account for about 70% of our training sample and brute-force attempts for 0.44%; in ToN-IoT [7] ransomware is 0.14% of flows. Accuracy is almost entirely determined by the frequent classes, while the rare classes (backdoors,

ransomware, man-in-the-middle, injection) are often the most damaging. Work on model compression in vision has shown that pruning and quantization degrade underrepresented classes disproportionately [8]. Whether distillation does the same to rare attacks in IDS has not, to our knowledge, been measured.

This paper measures it and tries to repair it. We distil an FT-Transformer teacher [9] into a two-layer multilayer perceptron (MLP) on three public IoT datasets and track per-class recall rather than accuracy alone. The findings are:

- **Distillation can erase rare attacks.** Standard KD and Decoupled KD (DKD) [10] leave the student 11.8–65.5 points below the teacher in minority-class recall. Part of this gap is capacity: a student trained with no teacher is already 10.7–11.5 points below the teacher. The additional loss caused by distillation is small on CIC-IoT2023 (0.3 points for KD) and large on Edge-IIoTset (14.9) and ToN-IoT (54.3).
- **A class-aware objective repairs the loss.** Minority-Aware KD (MA-KD) reweights the distillation term per class and distils from logits with the class prior removed [11]. It improves minority recall over KD on all three datasets. Ablation shows that sample-level class weighting and prior debiasing carry the gain; a per-class temperature schedule does not. We show that the debiasing step is a tempered form of the normalized target of Balanced Knowledge Distillation (BKD) [12] and compare the size of the two corrections on our class counts.
- **The teacher adds little beyond repair.** A student trained alone with class-weighted cross-entropy has higher macro-F1 than MA-KD on all three datasets and higher minority recall on two of them. We report this because it changes the practical advice: if a distillation pipeline is used, it must be class-aware, but at this model size a well-configured imbalance-aware loss is a strong and cheaper alternative.
- **The student is small.** It is a 21 KB ONNX file, needs about 0.01 MFLOPs per flow, and runs in about 14 μ s on one cloud CPU thread. We did not measure runtime memory or latency on physical IoT hardware.

We also attempted cross-dataset evaluation on a harmonized feature set and label taxonomy. It failed for every model, including the teacher. Because the harmonized schema already lowers in-domain performance, we report this as an unresolved problem and do not attribute it to the benchmarks alone. Figure 1 summarizes the main effect.

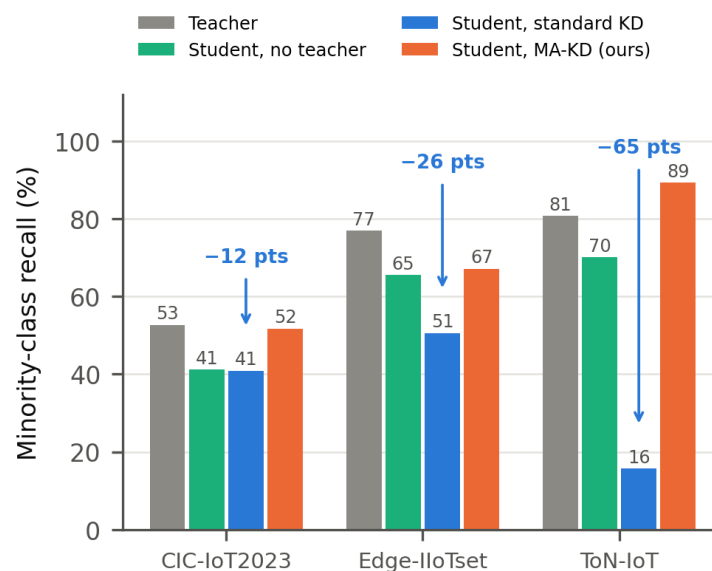


Figure 1. What distillation forgets. Mean recall on minority attack classes over three seeds. Standard KD (blue) loses 12, 26 and 65 points relative to its teacher and falls below a student trained with no teacher at all (green). MA-KD (orange) restores the lost recall.

Key findings. (i) Standard and decoupled KD make a small IoT detector worse than training it alone on Edge-IIoTset and ToN-IoT, where distillation itself removes 14.9 and 54.3 points of minority recall (Figure 1). (ii) Two class-aware changes to the distillation term, class weighting and prior debiasing, repair it (+10.8 to +73.5 points minority recall over KD on all three datasets). (iii) A class-weighted student without

a teacher is at least as good at this model size, so class awareness, not the teacher, is what protects rare attacks. (iv) The resulting 21 KB model classifies a flow in about 14 μ s on one cloud CPU thread.

The rest of the paper is organized as follows. Section 2 reviews related work. Section 3 defines the forgetting metrics and MA-KD and relates MA-KD to BKD. Section 4 describes the experimental protocol, Section 5 reports results, and Section 6 discusses implications and limitations.

2. Related Work

2.1. Lightweight and Distilled IoT Intrusion Detection

Compression of IDS models has been approached through feature selection, pruning, quantization and distillation. Wang et al. [13] combine KD with deep metric learning for industrial cyber-physical systems and reduce computational cost and model size by about 86% for a 0.4% accuracy loss. Yang et al. [5] apply self-distillation to network intrusion detection for resource-constrained devices. Wang et al. [14] distil a CNN-based detector and evaluate it on four public IDS datasets. Wang et al. [3] use self-knowledge distillation to build a compact IoT detector with multi-class accuracy above 99%. Xie et al. [15] distil two teachers into one student for industrial IoT, reducing model size about 250-fold. Wisanwanichthan and Thammawichai [4] report 92–95% parameter reduction across five IoT and UAV datasets. AL-Nomasy et al. [16] distil a transformer and add layer-wise explanations, and Yagiz and Goktas [17] combine distillation with a variational autoencoder and report 95.34% accuracy on Edge-IIoTset.

Table 1 summarizes these studies. Their headline results are aggregate measures (accuracy, overall detection or compression ratio). To our knowledge, none of them isolates how distillation changes recall on individual rare attack classes. That question is the subject of this paper. Because preprocessing, splits and class groupings differ between studies, we do not compare their reported numbers with ours; instead, we re-run representative KD objectives (standard KD and DKD) under one protocol and relate MA-KD to BKD analytically (Section 3.5).

Table 1. Distillation-based intrusion detection studies. Headline results as reported in each paper’s abstract; numbers are not comparable across rows because datasets, splits and class groupings differ.

Study	Distillation scheme	Datasets	Reported headline result	Evaluation focus
Wang et al. [13], 2022	KD + deep metric learning	NSL-KDD, CIC-IDS2017	$\approx$ 86% less compute and size; 0.4% accuracy loss	Accuracy, compression
Yang et al. [5], 2023	Self-KD	NIDS benchmarks	Fewer parameters and FLOPs than prior models	Accuracy, efficiency
Wang et al. [14], 2024	CNN teacher–student KD	Four public IDS datasets	Higher efficiency and accuracy than baselines	Accuracy, efficiency
Wang et al. [3], 2025	Self-KD (TBCLNN)	IoT IDS benchmarks	>99% multi-class accuracy	Accuracy, model size
Xie et al. [15], 2025	Dual-teacher KD	X-IIoTID, NSL-KDD, CICDDoS2019	$\approx$ 250 $\times$ smaller; 20 $\times$ fewer parameters	Accuracy, compression
Wisanwanichthan and Thammawichai [4], 2025	DNN KD	NSL-KDD, UNSW-NB15, CIC-IDS2017, IoTID20, UAV-IDS	92–95% fewer parameters; up to 6.12% better detection	Detection, compression
AL-Nomasy et al. [16], 2025	Transformer KD + explanations	IDS benchmarks	Better accuracy and efficiency than prior IDS	Accuracy, interpretability

Study	Distillation scheme	Datasets	Reported headline result	Evaluation focus
Yagiz and Goktas [17], 2025	KD + VAE	Edge-IIoTset, UKM-IDS20, CTU-13, NSL-KDD	95.34% accuracy on Edge-IIoTset	Accuracy, explainability
This work	Class-weighted KD from prior-debiased teacher logits	CIC-IoT2023, Edge-IIoTset, ToN-IoT	Minority recall +10.8 to +73.5 points over KD on 3/3 datasets; 21 KB file, about 14 μ s per flow on a cloud CPU	Per-class recall and precision, capacity vs. distillation gap, teacher-free baselines, cloud CPU latency

2.2. Compression, Distillation and Class Imbalance

Hooker et al. [8] showed that pruned image classifiers lose accuracy mainly on a small subset of hard, underrepresented examples, which headline accuracy conceals. In long-tailed recognition, class-balanced losses [18], focal loss [19], balanced softmax [20] and logit adjustment [11] correct the bias a model learns from a skewed prior. Balanced Knowledge Distillation (BKD) [12] combines an instance-balanced classification loss with a class-balanced distillation loss in which effective-number weights multiply the teacher’s class probabilities before renormalization. MA-KD also corrects the teacher’s class probabilities, through logit adjustment, and Section 3.5 shows that this correction is a tempered special case of the normalized BKD target. What MA-KD adds is a separate weight on each sample’s whole KL term, chosen by the sample’s label, and a correction strength that is set independently of the temperature. We also evaluate the method in an IDS setting against teacher-free imbalance baselines, which is where its limits appear.

2.3. Evaluation Pitfalls in ML-Based Security

Arp et al. [21] catalogue pitfalls in machine learning for security, including data snooping, spurious correlations and inappropriate metrics. Engelen et al. [22] document labeling and duplication problems in a widely used IDS dataset. Sarhan et al. [23] argue that incompatible feature sets across network IDS datasets prevent cross-dataset evaluation. We follow their advice by removing identifiers, de-duplicating before splitting, reporting per-class and macro metrics, and reporting paired differences with confidence intervals.

3. Problem Formulation and Method

3.1. Setting

Let $\{(x_i, y_i)\}$, $i = 1, \dots, n$, be flow records with $x_i \in \mathbb{R}^F$ and labels $y_i \in \{1, \dots, C\}$, and let n_c be the number of training samples in class c , with prior $\pi_c = n_c/n$. A teacher produces logits $z^T(x)$ and a student $z^S(x)$. We call class c a *minority* class if $\pi_c < 2\%$ and c is not the benign class. The set of minority classes is $\mathcal{M}$. Throughout, π is computed on the training split after de-duplication and sampling, not on the full collections; Section 4.1 reports how sampling changes it.

3.2. Measuring Forgetting

Let R_c denote test recall on class c . The minority recall of a model is the mean of R_c over $c \in \mathcal{M}$, written $\bar{R}_{\mathcal{M}}$. We define the *forgetting gap* (FG) of a student as

$$\text{FG} = \bar{R}_{\mathcal{M}}(\text{teacher}) - \bar{R}_{\mathcal{M}}(\text{student}) \quad (1)$$

in percentage points. A positive gap means the student has lost minority-class knowledge the teacher had; a negative gap means the student detects rare attacks better than the teacher. FG mixes two effects. A small student trained without any teacher (S-CE, written S_0) already loses minority recall relative to the teacher because of its limited capacity. We therefore split the gap as

$$\text{FG} = [\bar{R}_{\mathcal{M}}(\text{teacher}) - \bar{R}_{\mathcal{M}}(S_0)] + [\bar{R}_{\mathcal{M}}(S_0) - \bar{R}_{\mathcal{M}}(\text{student})] = G_{\text{cap}} + \Delta_{\text{dist}} \quad (2)$$

where G_{cap} is the capacity gap and Δ_{dist} is the change attributable to the training objective of the student. For a distilled student, a positive Δ_{dist} means distillation made the student detect fewer rare attacks than the same student trained without a teacher.

3.3. Why Standard KD Forgets

Standard KD [1] minimizes

$$\mathcal{L}_{\text{KD}} = \mathcal{L}_{\text{CE}}(z^S, y) + \lambda T^2 \text{KL}(\sigma(z^T/T) \parallel \sigma(z^S/T)) \quad (3)$$

A teacher trained on imbalanced data carries the class prior in its logits: for an ambiguous flow its probability mass leans toward frequent classes. Softening with $T > 1$ spreads that mass, and the Kullback–Leibler (KL) term, averaged over a batch where rare-class samples are a small fraction, pulls the student toward the prior. The student also has too little capacity to reproduce the teacher’s fine decision boundaries around small clusters, so it trades them away first. Both mechanisms point the same way.

3.4. Minority-Aware KD

MA-KD changes the distillation term in three ways (Figure 2).

Prior-debiased teacher targets. Following logit adjustment [11], the student distils from $\tilde{z}^T = z^T - \tau \log \pi$, which removes the teacher’s preference for frequent classes before softening. We use $\tau = 1$.

Class weights. Each sample’s KL term is weighted by w_y , where y is the label of that sample, with

$$w_c \propto \left(\frac{\max_k n_k}{n_c} \right)^\beta \quad (4)$$

capped at 50 and normalized so that the mean weight over training samples equals one.

Class temperatures. Each class receives $T_c = \max\{1, T_0(1 + \gamma H_c)\}$, where H_c is the teacher’s mean normalized entropy on training samples of class c . This was motivated by the idea that uncertain classes need different softening; Section 5.4 shows it has no measurable effect, and we recommend dropping it.

The full objective, for a batch of B samples, is

$$\mathcal{L}_{\text{MA}} = \mathcal{L}_{\text{CE}}(z^S, y) + \frac{\lambda}{B} \sum_{i=1}^B w_{y_i} T_{y_i}^2 \text{KL}(\sigma(\tilde{z}_i^T/T_{y_i}) \parallel \sigma(z_i^S/T_{y_i})) \quad (5)$$

Optional explanation consistency (XC). To encourage the student to rely on the same features as the teacher, we add

$$\mathcal{L}_{\text{XC}} = 1 - \cos(|x \odot \nabla_x z_y^S|, |x \odot \nabla_x z_y^T|) \quad (6)$$

weighted by μ . Teacher attributions are precomputed once on the training set.

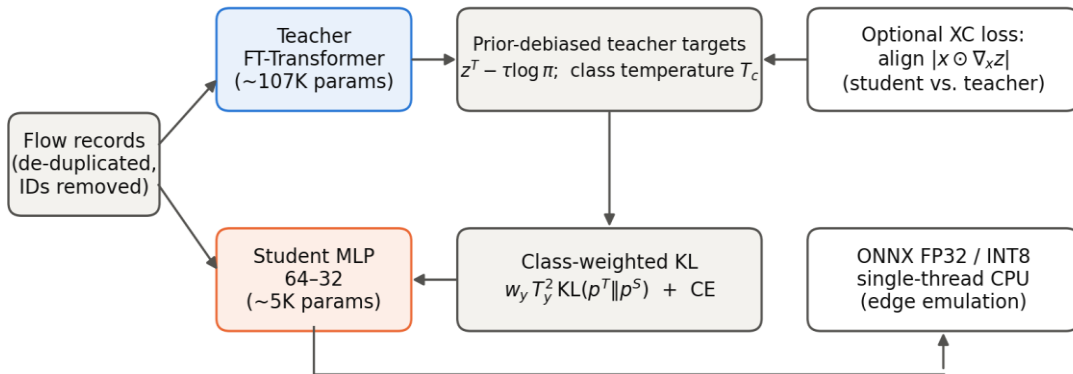


Figure 2. Overview of the pipeline. The teacher’s logits are debiased by the class prior before distillation; the per-sample KL term is weighted by class; an optional loss aligns gradient-based attributions. The trained student is exported to ONNX and benchmarked on one CPU thread.

3.5. Relation to Balanced Knowledge Distillation

The original BKD loss [12] is

$$\mathcal{L}_{\text{BKD}} = \mathcal{L}_{\text{CE}}(z^S, y) + T^2 \sum_c q_c \log \frac{q_c}{p_c^S}, \quad q_c = \frac{\omega_c \hat{p}_c^T}{\sum_k \omega_k \hat{p}_k^T}, \quad \omega_c = \frac{1-b}{1-b^{n_c}} \quad (7)$$

where $p^T = \text{softmax}(z^T/T)$, $p^S = \text{softmax}(z^S/T)$, n_c is the training count of class c and $b = 0.9999$. The weights ω_c act on the class components of the teacher distribution, and after renormalization the target q is

again a probability vector. The prior adjustment of MA-KD acts on the same components. Writing it in probability space gives

$$\tilde{p}_c^T = \text{softmax}\left(\frac{z^T - \tau \log \pi}{T}\right)_c = \frac{\hat{p}_c^T \pi_c^{-\tau/T}}{\sum_k \hat{p}_k^T \pi_k^{-\tau/T}} \quad (8)$$

so, the MA-KD target multiplies each teacher class probability by $\pi_c^{-\tau/T}$ and renormalizes. When ω_c is proportional to $1/n_c$, which the effective-number weights approach for classes with few samples, the normalized BKD target equals the MA-KD target with $\tau = T$. With our setting $\tau = 1$ and $T = 4$, MA-KD applies a weaker correction, $\pi_c^{-1/4}$. MA-KD therefore differs from BKD in two respects only. First, the strength of the class-probability correction is set by τ , independently of the temperature. Second, MA-KD multiplies each sample's entire KL term by $w_{y_{-i}}$, a weight chosen by the sample's ground-truth label, whereas BKD has no sample-level weight. The ablation row "KD + w_c " (Table 7) is this sample-level weighting alone; it is not BKD and was labelled "BKD-style" in our first submission, which was misleading.

How different the two corrections depend on the class counts. With $b = 0.9999$ the effective-number weights stop growing once a class has more than about 10^4 samples, so they do not follow $1/n_c$ for the large classes of our datasets. Table 2 compares, for the rarest and the largest class of each training split, the factor by which each method raises the rarest class relative to the largest. On CIC-IoT2023 the two corrections are of similar size (3.4 for BKD and 3.6 for MA-KD). On Edge-IIoTset and ToN-IoT the BKD weights correct about 4 and 12 times more strongly (20.2 against 5.1, and 55.4 against 4.6). We did not retrain the original BKD loss in this revision. The closest empirical analogue in our ablation is the debias-only row (KD + debias, Table 7), which applies the MA-KD correction without sample weights; a stronger correction of the BKD kind would be expected to move further toward the behaviour of logit-adjusted training (S-LA), with higher minority recall and more false alarms, but we have not measured this.

Table 2. Relative correction applied to the rarest class with respect to the largest class of each training split: effective-number weight ratio of BKD ($b = 0.9999$) and prior-adjustment ratio of MA-KD ($\pi^{-\tau/T}$ with $\tau = 1$, $T = 4$). The inverse-frequency ratio is shown for reference.

Dataset	Rarest class (n)	Largest class (n)	BKD ω ratio	MA-KD ratio	$n_{\max}/n_{\min}$
CIC-IoT2023	BruteForce (3,515)	DDoS (565,069)	3.37	3.56	160.8
Edge-IIoTset	MITM (508)	Normal (330,056)	20.19	5.05	649.7
ToN-IoT	ransomware (182)	normal (81,932)	55.43	4.61	450.2

4. Materials and Methods

4.1. Datasets and Preprocessing

We use CIC-IoT2023 [6] with its official eight-class grouping, the DNN release of Edge-IIoTset [24] with 15 classes, and the train_test_network release of ToN-IoT [7] with 10 classes. Table 3 summarizes them after preprocessing.

Preprocessing followed the recommendations in [21,22]. We removed IP addresses, ports, timestamps, URIs, raw payloads and the auxiliary binary label, which can act as shortcuts. We then removed exact duplicate rows over the remaining columns *before* splitting, so no identical record appears in both training and test sets. CIC-IoT2023 contains about 46 million rows; we sampled each class proportionally toward one million rows, while guaranteeing each class at least 5000 rows or all it had. This preserves the natural imbalance. Edge-IIoTset was sampled the same way toward 800,000 rows; ToN-IoT was used in full. The order of these steps differs by dataset: CIC-IoT2023 was sampled file by file and de-duplicated afterwards, because it does not fit in memory; Edge-IIoTset and ToN-IoT were de-duplicated first and then sampled. Each dataset was split 70/10/20 into training, validation and test sets, stratified by class. Numeric features received a signed log transform, $\text{sign}(x) \log(1 + |x|)$, then standardization with training statistics and clipping to $[-10, 10]$. Categorical columns were one-hot encoded over their ten most frequent training values. Constant columns were dropped.

Numeric columns were parsed with non-numeric entries set to missing. Missing values and positive or negative infinities were replaced by zero before the signed log transform. Categorical values were read as strings, so a missing entry forms its own category when it is among the ten most frequent values. The model inputs are all columns of the released files except those removed above: the 46 numeric flow features

of CIC-IoT2023 (44 non-constant in the training split); the remaining packet fields of Edge-IIoTset, of which seven are categorical (HTTP method, referer and version, DNS query-name length, MQTT connect-acknowledge flags, protocol name and topic), giving 39 non-constant inputs after encoding; and the remaining Zeek connection, DNS, SSL, HTTP and weird-event fields of ToN-IoT, giving 107 inputs after one-hot encoding of protocol, service, connection state and the categorical DNS, SSL and HTTP fields. The encoded feature names are listed in the released code.

The 5000-record floor raises the share of the rarest classes in the training split relative to the de-duplicated collection, and the proportional sampling of CIC-IoT2023 lowers the share of the largest ones. Because MA-KD, S-WCE and S-LA compute priors and weights from the training split, this changes their effective correction. ToN-IoT was not sampled, so its training prior equals the de-duplicated prior. In CIC-IoT2023 and Edge-IIoTset, classes that fell below the floor were kept in full while the large classes were sampled, so the rare classes are over-represented in the training split by roughly the inverse of the sampling fraction of the large classes; in CIC-IoT2023 this applies to brute force, whose retained records sit at the floor.

Partitions were drawn at random at the record level. Records from the same capture session, attack run or device can therefore appear in the training, validation and test sets, and removing identical rows does not remove this dependence. CIC-IoT2023 releases no capture or device identifier with its flow records, and we did not use the host addresses of the other two datasets to group records. We did not build grouped or temporal splits. All results in this paper therefore describe held-out records from the same three collections; they are not evidence of generalization to new captures, devices or networks.

Table 3. Datasets after de-duplication, sampling and encoding. Minority classes are non-benign classes with less than 2% of the training set.

Dataset	Features	Classes	Train	Test	Largest (%)	Smallest (%)	Minority classes
CIC-IoT2023	44	8	805,927	230,265	70.1	0.44	Recon, Spoofing, BruteForce DDoS_HTTP, Vulnerability_scanner, SQL_injection, Uploading, Backdoor, XSS, Port_Scanning, Ransomware, Fingerprinting, MITM
Edge-IIoTset	39	15	590,219	168,635	55.9	0.09	backdoor, mitm, ransomware
ToN-IoT	107	10	130,935	37,410	62.6	0.14	

4.2. Models

The teacher is an FT-Transformer [9] with a per-feature tokenizer, three pre-norm encoder layers of width 64 with 8 heads, and a CLS token (107–115K parameters). Each encoder layer has a feed-forward block of width 128 with GELU activation and dropout 0.1; a final layer normalization and a linear layer act on the CLS token. The teacher was trained with unweighted cross-entropy and no label smoothing, once per dataset, with AdamW [25], learning rate 5×10^{-4} , batch size 1024, a one-cycle schedule, mixed precision and early stopping on validation macro-F1 (at most 25 epochs, patience 5).

The student is an MLP with hidden layers of 64 and 32 units, ReLU and dropout 0.1 (5.1–9.3K parameters, depending on input width). Every student was trained with AdamW, learning rate 2×10^{-3} , weight decay 10^{-4} , batch size 2048, a one-cycle schedule and early stopping on validation macro-F1 (at most 40 epochs, patience 8). Each student configuration was run with seeds 0, 1 and 2.

Loss conventions. All losses are averaged over the mini-batch. Cross-entropy uses the mean reduction; with class weights (S-WCE, KD+WCE) this is the weighted mean $\sum_i w_{y_{[i]}} \ell_i / \sum_i w_{y_{[i]}}$. The KL term is summed over classes for each sample, multiplied by T^2 , and averaged over the B samples of the batch. In MA-KD the weights $w_{y_{[i]}}$ multiply the per-sample KL terms before this average, and the sum is divided by B , not by $\sum_i w_{y_{[i]}}$, so the class weights also scale the distillation term as a whole; their normalization to a mean of one over the training set keeps its expected size equal to that of standard KD. The DKD term is a batch mean of their per-sample values.

4.3. Compared Methods

We compare three groups of students with identical architecture and training budget:

- Teacher-free: cross-entropy (S-CE); class-weighted cross-entropy with the weights w_c above (S-WCE); logit-adjusted cross-entropy [11] (S-LA), which minimizes $\text{CE}(z^S + \tau \log \pi, y)$ with $\tau = 1$ during training and predicts with $\arg\max z^S$ at test time.
- Standard distillation: KD [1] with $T = 4$, $\lambda = 1$; DKD [10] with $\alpha = 1$, $\beta = 8$, $T = 4$; KD with a class-weighted cross-entropy term (KD+WCE). KD was run with this single setting. We did not tune its temperature or weight, and all statements about the deterioration caused by KD refer to $T = 4$ and $\lambda = 1$.
- Proposed: MA-KD with $T_0 = 4$, $\beta = 0.5$, $\gamma = -0.5$, $\lambda = 1$, $\tau = 1$; MA-KD+XC with $\mu = 1$; and the ablations in Table 7.

Hyperparameters were fixed before testing and shared across datasets. They were not tuned per dataset for any method, including MA-KD.

4.4. Metrics and Statistics

We report accuracy, macro-F1, balanced accuracy, minority recall and the forgetting gap on the held-out test set as mean $\pm$ standard deviation over three student seeds. Within a dataset the three seeds share one teacher and one data split and differ only in student initialization and batch order, so they measure run-to-run variation rather than independent evidence. The unit of evidence for claims across benchmarks is therefore the dataset. For each pair of methods, we report the difference of seed means on each dataset with a 95% t-interval built from the two seed means and standard deviations (three seeds each, 4 degrees of freedom). This interval does not use the pairing of runs by seed; it describes run-to-run variation with the teacher and the split held fixed. The test records that only one of two methods classifies correctly, overall and for the minority classes, with the exact McNemar test [26]. For macro-F1 we also give 95% bootstrap intervals over test records [27]. With three datasets, the smallest exact two-sided p-value of the signed-rank test [28] is 0.25 even when all three differences share a sign, so we report the direction and size of the differences and do not claim significance across datasets. The McNemar tests treat test records as the unit and describe sampling variation within one dataset only.

4.5. Deployment Emulation

No physical edge device was available. We therefore exported models to ONNX, applied dynamic INT8 weight quantization [29], and measured inference with ONNX Runtime [30-31] restricted to one intra-op thread on a Kaggle CPU instance. Latency is the median over 1000 single-flow calls (300 for the teacher) after 50 warm-up calls; throughput uses batches of 256. These measurements give file size and single-thread latency on a cloud CPU. They do not measure runtime memory, energy, or latency on gateway or microcontroller hardware.

5. Results

5.1. Distillation Forgets Rare Attacks

Table 4 gives the main results and Figure 3 shows per-class recall. Standard KD leaves the frequent classes nearly untouched: DDoS, Mirai, the DDoS classes of Edge-IIoTset and normal traffic stay within about one point of the teacher. The losses fall on the rare classes. On ToN-IoT the KD student's recall drops from 69.2% to 0% on ransomware, from 74.5% to 13.1% on man-in-the-middle and from 98.7% to 34.1% on backdoor, while its accuracy stays at 95.7%. On Edge-IIoTset, SQL injection falls from 64.0% to 4.7%, XSS from 75.9% to 6.6% and uploading from 60.5% to 3.5%. On CIC-IoT2023, brute force falls from 40.2% to 14.2%.

The forgetting gap of KD (11.8, 26.4 and 65.1 points) is not caused by distillation alone. Table 5 splits it with Eq. (2). The capacity gap between the teacher and the student trained without a teacher is similar on the three datasets (11.4, 11.5 and 10.7 points). The distillation-specific change differs widely. On CIC-IoT2023, KD loses only 0.3 points of minority recall relative to S-CE, so almost the whole teacher-student gap there is a capacity effect; S-CE itself recalls brute force at only 15.3%. On Edge-IIoTset and ToN-IoT, KD loses 14.9 and 54.3 points relative to S-CE, and DKD loses 29.3 and 54.8. In macro-F1 the distillation-specific loss of KD is 0.9, 11.9 and 17.2 points. Our claim of rare-class deterioration caused by distillation therefore holds for Edge-IIoTset and ToN-IoT and not for CIC-IoT2023, and it holds for the KD setting we tested ($T = 4$, $\lambda = 1$); a different temperature or weight might reduce it.

Table 4. Test results (% , mean $\pm$ std over three seeds). FG = forgetting gap in minority recall relative to the teacher (points; negative means the student exceeds the teacher). Bold: best student per dataset and column.

Method	Params	Accuracy	Macro-F1	Balanced acc.	Minority recall	FG
<i>CIC-IoT2023</i>						
Teacher (FT-Transformer)	106,760	90.23	77.21	75.18	52.71	0.00
S-CE	5,224	89.46 $\pm$ 0.04	72.43 $\pm$ 0.18	70.31 $\pm$ 0.14	41.28 $\pm$ 0.53	11.44
S-WCE	5,224	88.84 $\pm$ 0.03	74.98 $\pm$ 0.12	74.24 $\pm$ 0.18	51.31 $\pm$ 0.59	1.41
S-LA	5,224	83.57 $\pm$ 0.41	71.23 $\pm$ 0.34	74.45 $\pm$ 0.33	54.03 $\pm$ 0.91	-1.32
KD	5,224	89.31 $\pm$ 0.03	71.55 $\pm$ 0.08	69.69 $\pm$ 0.06	40.94 $\pm$ 0.24	11.77
DKD	5,224	88.90 $\pm$ 0.08	68.58 $\pm$ 1.43	67.35 $\pm$ 0.84	38.06 $\pm$ 2.22	14.65
KD+WCE	5,224	89.22 $\pm$ 0.07	72.89 $\pm$ 0.22	71.22 $\pm$ 0.26	44.60 $\pm$ 1.39	8.11
MA-KD	5,224	88.98 $\pm$ 0.04	73.52 $\pm$ 0.17	73.37 $\pm$ 0.22	51.78 $\pm$ 0.58	0.94
MA-KD+XC	5,224	88.87 $\pm$ 0.08	73.21 $\pm$ 0.21	72.96 $\pm$ 0.41	50.96 $\pm$ 1.53	1.75
<i>Edge-IIoTset</i>						
Teacher (FT-Transformer)	106,575	97.17	87.26	82.85	77.00	0.00
S-CE	5,135	96.43 $\pm$ 0.12	79.96 $\pm$ 0.34	75.03 $\pm$ 0.44	65.48 $\pm$ 0.51	11.51
S-WCE	5,135	96.63 $\pm$ 0.11	81.90 $\pm$ 0.36	78.68 $\pm$ 0.43	70.97 $\pm$ 0.49	6.03
S-LA	5,135	87.68 $\pm$ 0.11	74.82 $\pm$ 0.47	80.51 $\pm$ 0.47	74.30 $\pm$ 0.57	2.70
KD	5,135	95.56 $\pm$ 0.10	68.08 $\pm$ 0.99	65.08 $\pm$ 0.58	50.61 $\pm$ 0.73	26.39
DKD	5,135	94.84 $\pm$ 0.06	54.25 $\pm$ 0.31	54.60 $\pm$ 0.29	36.14 $\pm$ 0.48	40.86
KD+WCE	5,135	96.04 $\pm$ 0.18	75.20 $\pm$ 1.34	71.60 $\pm$ 1.50	60.40 $\pm$ 2.13	16.60
MA-KD	5,135	96.36 $\pm$ 0.14	79.22 $\pm$ 0.90	76.05 $\pm$ 0.52	67.18 $\pm$ 0.61	9.81
MA-KD+XC	5,135	96.23 $\pm$ 0.09	79.03 $\pm$ 0.56	75.47 $\pm$ 0.41	66.51 $\pm$ 0.56	10.49
<i>ToN-IoT</i>						
Teacher (FT-Transformer)	114,954	97.89	90.33	91.65	80.83	0.00
S-CE	9,322	97.09 $\pm$ 0.03	87.61 $\pm$ 0.25	87.51 $\pm$ 0.50	70.09 $\pm$ 1.77	10.73
S-WCE	9,322	96.59 $\pm$ 0.34	87.69 $\pm$ 0.46	93.72 $\pm$ 0.20	90.68 $\pm$ 0.46	-9.86
S-LA	9,322	94.01 $\pm$ 0.27	78.92 $\pm$ 0.44	93.41 $\pm$ 0.19	95.25 $\pm$ 0.40	-14.42
KD	9,322	95.72 $\pm$ 0.39	70.38 $\pm$ 3.28	70.16 $\pm$ 2.79	15.75 $\pm$ 8.71	65.08
DKD	9,322	94.53 $\pm$ 0.42	66.95 $\pm$ 2.17	67.06 $\pm$ 2.59	15.33 $\pm$ 8.05	65.49
KD+WCE	9,322	96.61 $\pm$ 0.09	88.04 $\pm$ 0.05	91.61 $\pm$ 0.16	86.25 $\pm$ 0.72	-5.42
MA-KD	9,322	96.20 $\pm$ 0.24	86.00 $\pm$ 0.99	92.49 $\pm$ 0.34	89.27 $\pm$ 0.64	-8.44
MA-KD+XC	9,322	94.84 $\pm$ 0.96	81.82 $\pm$ 2.37	89.98 $\pm$ 1.78	89.26 $\pm$ 0.60	-8.43

Table 5. Decomposition of the forgetting gap in minority recall (points) into the capacity gap G_{cap} (teacher minus S-CE) and the distillation-specific change Δ_{dist} (S-CE minus the student), following Eq. (2). Means over three seeds.

Student	Teacher	S-CE (S_0)	Student	G_{cap}	Δ_{dist}	FG
CIC-IoT2023						
KD	52.71	41.28	40.94	11.43	+0.34	11.77
DKD	52.71	41.28	38.06	11.43	+3.22	14.65
MA-KD	52.71	41.28	51.78	11.43	-10.50	0.93
S-WCE	52.71	41.28	51.31	11.43	-10.03	1.40
Edge-IIoTset						
KD	77.00	65.48	50.61	11.52	+14.87	26.39
DKD	77.00	65.48	36.14	11.52	+29.34	40.86
MA-KD	77.00	65.48	67.18	11.52	-1.70	9.82
S-WCE	77.00	65.48	70.97	11.52	-5.49	6.03
ToN-IoT						
KD	80.83	70.09	15.75	10.74	+54.34	65.08
DKD	80.83	70.09	15.33	10.74	+54.76	65.50
MA-KD	80.83	70.09	89.27	10.74	-19.18	-8.44
S-WCE	80.83	70.09	90.68	10.74	-20.59	-9.85

For MA-KD and S-WCE a negative Δ_{dist} means the student detects more rare attacks than S-CE. S-WCE uses no teacher and is shown for reference.

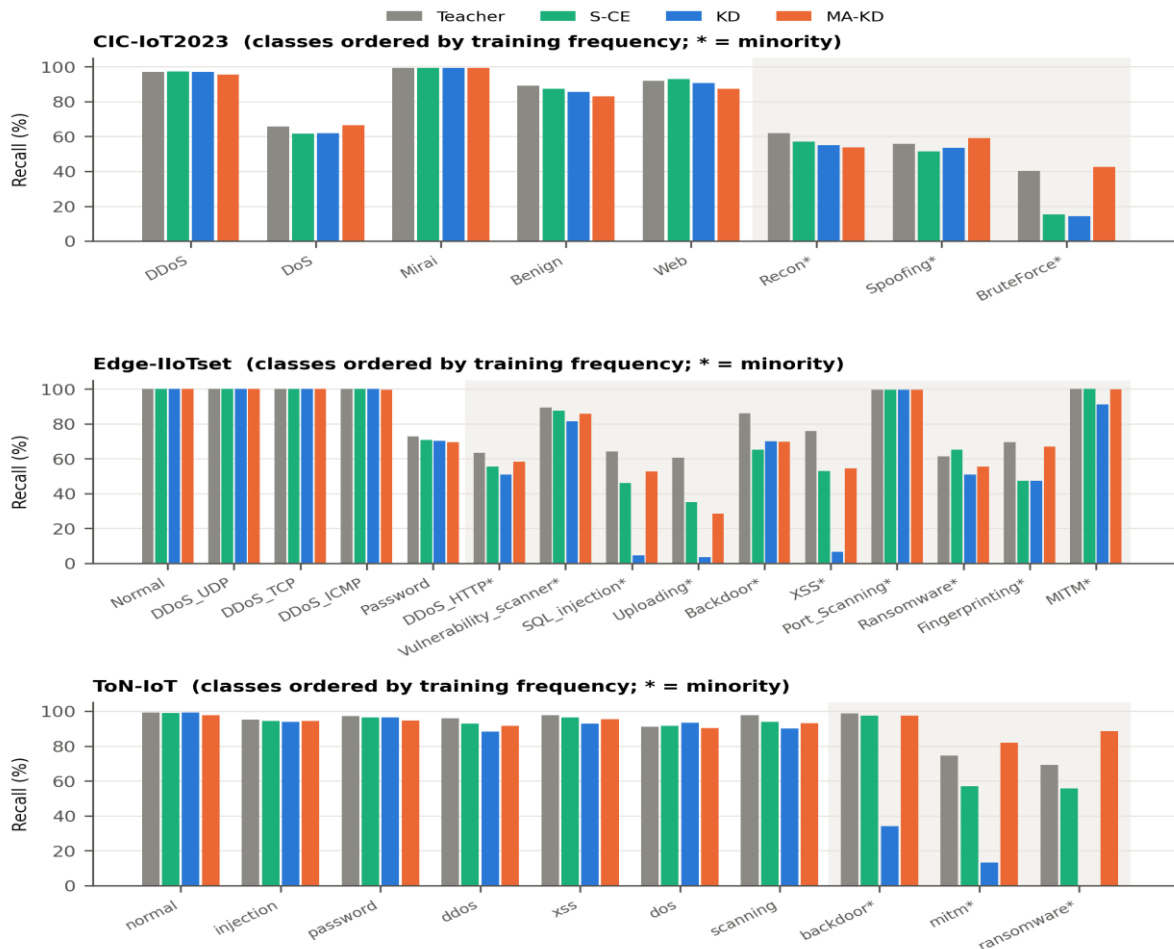


Figure 3. Per-class test recall of the teacher, a teacher-free student (S-CE), a KD student and an MA-KD student (mean over three seeds). Classes are ordered by training frequency; the shaded region marks minority classes.

5.2. MA-KD Repairs the Forgetting

MA-KD restores most of what KD loses. Its forgetting gaps are 0.9, 9.8 and -8.4 points; its distillation-specific change is -10.5 , -1.7 and -19.2 points, so on all three datasets MA-KD detects more rare attacks than the student trained without a teacher. On ToN-IoT the student exceeds the teacher on rare classes, reaching 88.5% on ransomware and 81.9% on man-in-the-middle. Macro-F1 rises by 2.0, 11.1 and 15.6 points over KD. Table 6 lists the dataset-level paired differences. MA-KD is ahead of KD and DKD in minority recall and macro-F1 on all three datasets. For MA-KD minus KD, the 95% intervals of the minority-recall difference are [9.6, 12.1], [14.7, 18.4] and [56.4, 90.7] points, and none of the six intervals for KD and DKD includes zero. As explained in Section 4.4, three datasets cannot yield a small p -value for a test across datasets (exact $p = 0.25$ when all three differences are positive), so the evidence across benchmarks is the consistency of direction and size, which we do not overstate. The bootstrap intervals of MA-KD and KD macro-F1 do not overlap on any dataset (for example, 79.6 [78.9, 80.4] vs. 68.3 [67.6, 69.0] on Edge-IIoTset). Exact McNemar tests find the MA-KD and KD predictions significantly different in every seed ($p < 0.001$), and on minority-class samples MA-KD corrects far more KD errors than it introduces (on ToN-IoT, 304–549 against none); on CIC-IoT2023 MA-KD makes slightly more errors overall (accuracy 88.98% vs. 89.31%) while correcting about twice as many minority samples as it loses.

5.3. Comparison with Teacher-Free Students

The teacher-free baselines change the picture (Tables 4 and 6, Figure 4). Against class-weighted cross-entropy (S-WCE), the minority recall of MA-KD is 0.5 points higher on CIC-IoT2023 and 3.8 and 1.4 points lower on Edge-IIoTset and ToN-IoT, and its macro-F1 is lower on all three datasets (by 1.5, 2.7 and 1.7 points). The 95% intervals of the minority-recall difference are $[-1.2, 2.1]$, $[-5.3, -2.3]$ and $[-3.0, 0.1]$ points: the two methods cannot be told apart on CIC-IoT2023 and ToN-IoT, and S-WCE is better on Edge-IIoTset. We make no claim of equivalence; on these data S-WCE is at least as good as MA-KD. On these two datasets S-WCE corrects more of the rare-attack errors of MA-KD than the reverse. The per-class predictions of S-WCE were not part of the per-class analysis of our experiments, so Figure 3 and the false-alarm figures below do not include it; its accuracy (88.8%, 96.6% and 96.6%) is within 0.4 points of MA-KD on every dataset.

Class awareness has a false-alarm cost that accuracy hides. The benign false-positive rate of MA-KD is 17.0% on CIC-IoT2023 (14.3% for KD and 12.7% for S-CE), 0.02% on Edge-IIoTset and 2.2% on ToN-IoT (0.8% for KD); for S-LA it is 16.8%, 17.8% and 5.2%.

Against logit-adjusted cross-entropy (S-LA), MA-KD has 2.3, 7.1 and 6.0 points lower minority recall but higher macro-F1 on every dataset (by 2.3, 4.4 and 7.1 points). S-LA buys its recall with false alarms: on Edge-IIoTset it classifies only 82.2% of normal flows correctly, against 99.98% for MA-KD, and its accuracy drops to 87.7%. Against plain S-CE, MA-KD gains 10.5, 1.7 and 19.2 points of minority recall with macro-F1 differences of +1.1, -0.7 and -1.6 points.

These results limit what can be claimed for the teacher. MA-KD removes the harm that standard distillation causes, but at a student size of a few thousand parameters the teacher's soft labels do not carry information that a class-weighted loss on hard labels fails to provide.

Table 6. Dataset-level differences, MA-KD minus each baseline (points; difference of seed means, with 95% t-interval over seeds). Replaces the pooled signed-rank table of the first submission.

Baseline	Metric	CIC-IoT2023	Edge-IIoTset	ToN-IoT
KD	MinR	+10.84 [9.61, 12.07]	+16.57 [14.70, 18.44]	+73.52 [56.37, 90.67]
	F1	+1.97 [1.60, 2.34]	+11.14 [8.51, 13.77]	+15.62 [8.89, 22.35]
DKD	MinR	+13.72 [9.22, 18.22]	+31.04 [29.52, 32.56]	+73.94 [58.09, 89.79]
	F1	+4.94 [2.11, 7.77]	+24.97 [23.10, 26.84]	+19.05 [14.37, 23.73]
KD+WCE	MinR	+7.18 [4.22, 10.14]	+6.78 [2.43, 11.13]	+3.02 [1.13, 4.91]
	F1	+0.63 [0.08, 1.18]	+4.02 [0.85, 7.19]	$-2.04 [-3.99, -0.09]$
S-CE	MinR	+10.50 [8.96, 12.04]	+1.70 [0.14, 3.26]	+19.18 [15.48, 22.88]
	F1	+1.09 [0.60, 1.58]	$-0.74 [-2.63, 1.15]$	$-1.61 [-3.61, 0.39]$
S-WCE	MinR	+0.47 $[-1.15, 2.09]$	$-3.79 [-5.33, -2.25]$	$-1.41 [-2.96, 0.14]$

Baseline	Metric	CIC-IoT2023	Edge-IIoTset	ToN-IoT
S-LA	F1	-1.46 [-1.87, -1.05]	-2.68 [-4.58, -0.78]	-1.69 [-3.83, 0.45]
	MinR	-2.25 [-4.37, -0.13]	-7.12 [-8.76, -5.48]	-5.98 [-7.46, -4.50]
	F1	+2.29 [1.54, 3.04]	+4.40 [2.41, 6.39]	+7.08 [4.95, 9.21]

MinR = minority recall; F1 = macro-F1.

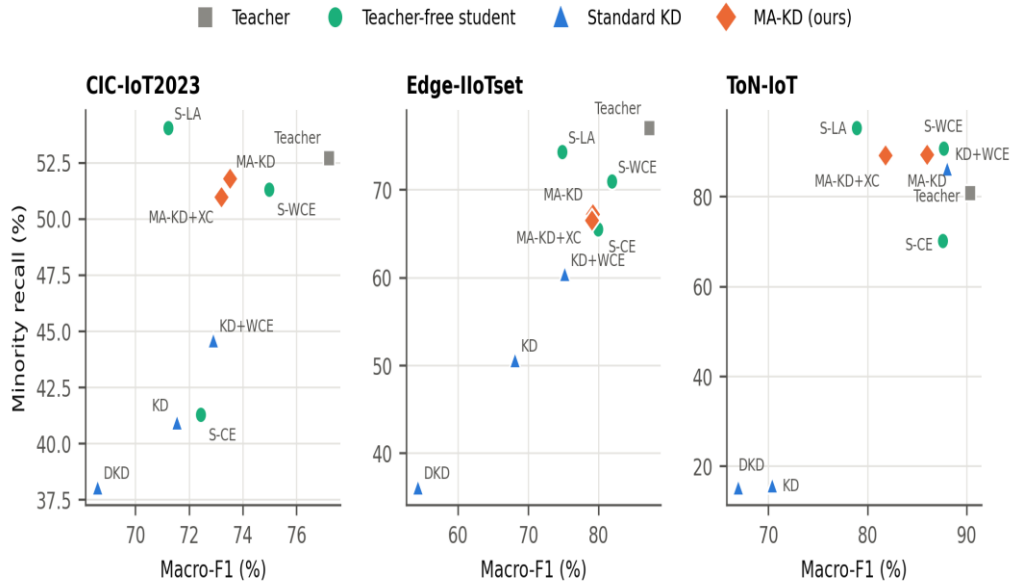


Figure 4. Macro-F1 versus minority recall for all students and the teacher (mean over three seeds). Standard KD students (triangles) lie below and to the left of the teacher-free students (circles); MA-KD (diamonds) recovers their region but does not move beyond it.

5.4. Ablation

Table 7 adds the MA-KD components one at a time. Per-class temperatures alone behave exactly like KD (for example, 8.6% minority recall on ToN-IoT against 15.8% for KD). Sample-level class weights alone (KD + w_c), which weight each sample's whole KL term and are not the BKD loss (Section 3.5), give most of the gain on Edge-IIoTset (62.3%). Prior debiasing alone gives most of the gain on ToN-IoT (83.3%). On CIC-IoT2023 neither is enough by itself (43.0% and 42.1%), but together they reach 50.8%. Combining weights and debiasing reaches 50.8, 66.7 and 88.9% minority recall on the three datasets; adding temperatures changes these by less than one point, within seed variation. Compared with sample-level weighting alone, full MA-KD raises minority recall by 8.8, 4.9 and 19.1 points, with macro-F1 changes of +1.1, +2.2 and 0.0 points, which shows the value of debiasing the teacher on top of class weighting. The effective method is therefore class-weighted distillation from prior-debiased teacher logits, and the temperature schedule can be removed.

Table 7. Ablation of MA-KD components (% , mean over three seeds). w_c : sample-level class weights; T_c : class temperatures; debias: prior-debiased teacher logits.

Variant	F1 CIC	F1 Edge	F1 ToN	MinR CIC	MinR Edge	MinR ToN
KD	71.55	68.08	70.38	40.94	50.61	15.75
KD + w_c (sample-level weights)	72.47	77.03	86.03	43.02	62.28	70.14
KD + T_c	71.53	67.82	67.89	40.73	50.46	8.60
KD + debias	71.66	67.65	87.09	42.09	50.99	83.27
KD + w_c + debias	73.33	78.85	85.10	50.79	66.69	88.89
MA-KD (all three)	73.52	79.22	86.00	51.78	67.18	89.27

Variant	F1 CIC	F1 Edge	F1 ToN	MinR CIC	MinR Edge	MinR ToN
MA-KD + XC	73.21	79.03	81.82	50.96	66.51	89.26

F1 = macro-F1; MinR = minority recall.

5.5. Sensitivity

Figure 5 varies β and γ on ToN-IoT. The temperature parameter γ has little effect at any β , in line with the ablation. The class-weight exponent β controls a clear trade-off: raising it from 0 to 1 increases minority recall from about 82–83% to about 90% while macro-F1 falls from about 87.5% to about 84%, because the student starts predicting rare classes for some frequent-class flows. β is therefore an operating-point choice. A deployment that values catching rare attacks over false alarms should use a higher value.

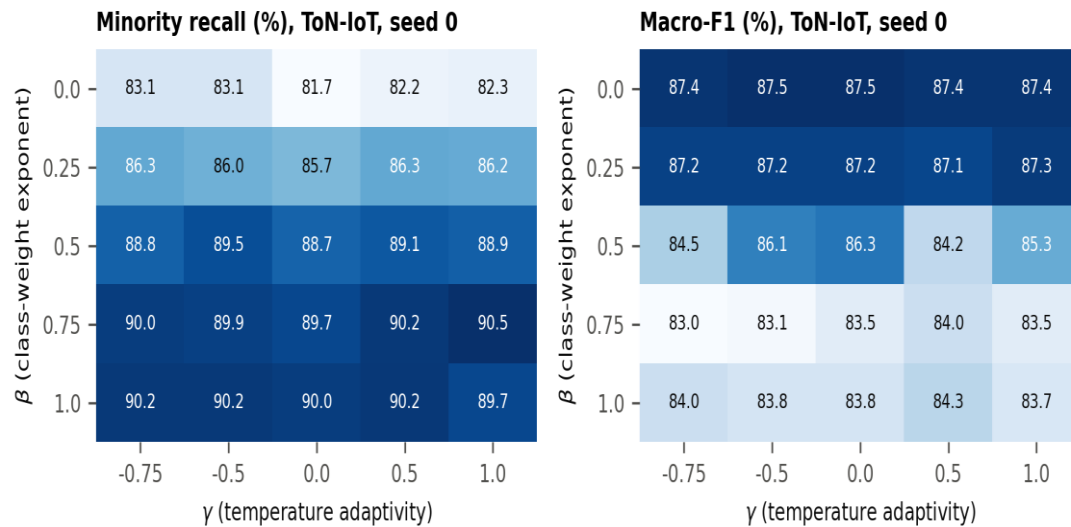


Figure 5. Sensitivity of MA-KD to the class-weight exponent β and the temperature parameter γ on ToN-IoT (seed 0).

5.6. Explanation Consistency (Exploratory)

Table 8 compares Integrated Gradients [32] attributions of each student with those of the teacher on 1000 test flows. Over all test flows, the XC loss raises agreement on CIC-IoT2023 (Spearman 0.690 to 0.744, top-5 Jaccard 0.401 to 0.471) and Edge-IIoTset (0.807 to 0.833; 0.408 to 0.626) and changes little on ToN-IoT (0.767 to 0.777; 0.372 to 0.359). On minority flows the results are mixed: top-5 overlap rises on Edge-IIoTset (0.510 to 0.607) but falls on CIC-IoT2023 (0.438 to 0.348) and ToN-IoT (0.413 to 0.363), and the rank correlations change by less than 0.02. XC also lowers macro-F1 on ToN-IoT from 86.0% to 81.8% (Table 4). Agreement with the teacher's attributions measures similarity between two models. It does not show that either attribution is faithful to the model's decision process or useful to an analyst, and we make no auditability claim.

Table 8. Agreement between student and teacher Integrated-Gradients attributions on 1000 test flows (seed 0). ρ : Spearman rank correlation; J: Jaccard overlap of the five most important features.

Dataset	Student	ρ (all)	Top-5 J (all)	ρ (minority)	Top-5 J (minority)
CIC-IoT2023	S-CE	0.655	0.381	0.787	0.402
CIC-IoT2023	KD	0.679	0.354	0.768	0.495
CIC-IoT2023	MA-KD	0.690	0.401	0.771	0.438
CIC-IoT2023	MA-KD+XC	0.744	0.471	0.786	0.348
Edge-IIoTset	S-CE	0.800	0.442	0.797	0.454
Edge-IIoTset	KD	0.791	0.456	0.786	0.479
Edge-IIoTset	MA-KD	0.807	0.408	0.785	0.510

Dataset	Student	q (all)	Top-5 J (all)	q (minority)	Top-5 J (minority)
Edge-IIoTset	MA-KD+XC	0.833	0.626	0.798	0.607
ToN-IoT	S-CE	0.816	0.354	0.793	0.271
ToN-IoT	KD	0.771	0.309	0.782	0.278
ToN-IoT	MA-KD	0.767	0.372	0.755	0.413
ToN-IoT	MA-KD+XC	0.777	0.359	0.757	0.363

5.7. Deployment Cost

Table 9. Inference cost on one cloud CPU thread with ONNX Runtime (seed-0 models). Latency: median per single flow. Throughput: batches of 256.

Model	Precision	Params	MFLOPs	Size (KB)	Latency (ms)	Throughput (10 ³ /s)	Macro-F1 (%)
<i>CIC-IoT2023</i>							
Teacher	FP32	106,760	10.409	490.7	0.6494	1.7	77.23
Teacher	INT8	106,760	10.409	245.1	0.6363	1.7	77.17
Student (MA-KD)	FP32	5,224	0.010	21.2	0.0139	4,183.5	73.71
Student (MA-KD)	INT8	5,224	0.010	9.1	0.0187	3,159.7	73.12
<i>Edge-IIoTset</i>							
Teacher	FP32	106,575	9.100	489.9	0.5785	2.1	87.25
Teacher	INT8	106,575	9.100	243.1	0.5704	1.8	87.37
Student (MA-KD)	FP32	5,135	0.010	20.8	0.0142	4,150.9	79.62
Student (MA-KD)	INT8	5,135	0.010	9.0	0.0188	3,175.9	77.25
<i>ToN-IoT</i>							
Teacher	FP32	114,954	30.207	522.7	1.5011	0.5	90.33
Teacher	INT8	114,954	30.207	276.8	1.4492	0.5	89.98
Student (MA-KD)	FP32	9,322	0.018	37.2	0.0146	2,709.6	86.08
Student (MA-KD)	INT8	9,322	0.018	13.1	0.0190	2,150.1	79.43

Table 9 reports inference cost. The MA-KD student has 12–21× fewer parameters and about 900–1600× fewer FLOPs than the teacher. As an ONNX FP32 file it occupies 21–37 KB and answers a single flow in about 14 μs on one cloud CPU thread, 41–103× faster than the teacher, with a throughput of 2.7–4.2 million flows per second in batches. Dynamic INT8 quantization shrinks the student to 9–13 KB but does not make it faster at this size, and it costs 0.6, 2.4 and 6.7 macro-F1 points. We therefore prefer FP32 for students this small. These figures describe relative cost between teacher and student under ONNX Runtime. They do not measure peak runtime memory, which includes the runtime itself, or latency and energy on constrained devices, and they do not show that the student runs on any particular IoT hardware.

5.8. Cross-Dataset Evaluation

To test transfer across networks, we mapped all three datasets to 13 features that can be computed from each and to a common seven-class taxonomy. Table 10 gives the source of every harmonized feature in each dataset and Table 11 the class mapping. The correspondence is imperfect. CIC-IoT2023 flag features are rates over a window of packets, ToN-IoT flags are derived from the Zeek connection state, and Edge-IIoTset flags come from single packets. Edge-IIoTset has no flow duration or packet count, so the packet count is constant and duration is taken from the UDP inter-arrival field; packet size is the total packet size in CIC-IoT2023, the mean IP bytes per packet in ToN-IoT and the TCP payload length in Edge-IIoTset.

Teachers and students were retrained on this schema and tested on other datasets pairwise and in leave-one-dataset-out form. Table 12 reports every direction.

Even in-domain, the harmonized features support only 41–78% macro-F1, far below the native feature sets. Across datasets, seven-class macro-F1 lies between 0.7% and 25.3% for every direction and model, and the binary detection rates that look higher come with false-positive rates of 57–84% on average. We cannot attribute this failure to the benchmarks alone. In-domain performance already drops after the reduction to 13 features, so part of the loss is information removed by the mapping; the mapped features do not mean the same thing in each dataset; and the datasets differ in traffic, tooling and labelling. Our design does not separate these explanations. What the experiment does show is that no compared model, teacher included, transfers under this schema, so the distillation comparison cannot be extended to new networks with it. Sarhan et al. [23] argue that a common feature extractor applied to raw traffic is needed for such evaluation.

Table 10. Source of the 13 harmonized features in each dataset.

Feature	CIC-IoT2023	ToN-IoT	Edge-IIoTset
is_tcp	TCP	proto = tcp	tcp.flags, tcp.seq, tcp.ack or tcp.len non-zero
is_udp	UDP	proto = udp	udp.port, udp.stream or udp.time_delta non-zero
is_icmp	ICMP	proto = icmp	icmp.checksum or icmp.seq_le non-zero
svc_http	HTTP	service contains http	http.request.method, http.response or http.content_length present
svc_https	HTTPS	service = ssl	http.tls_port non-zero
svc_dns	DNS	service = dns	dns.qry.type or dns.qry.name.len present
flag_syn	syn_flag_number	TCP and conn_state $\neq$ OTH	tcp.connection.syn or synack
flag_ack	ack_flag_number	conn_state $\in$ {S1, SF, S2, S3, RSTO, RSTR}	tcp.flags.ack
flag_fin	fin_flag_number	conn_state $\in$ {SF, S2, S3, SH, SHR}	tcp.connection.fin
flag_rst	rst_flag_number	conn_state = REJ or RST	tcp.connection.rst
log_pkt_size	log(1 + Tot size)	log(1 + IP bytes / packets)	log(1 + tcp.len)
log_duration	log(1 + flow_duration)	log(1 + duration)	log(1 + udp.time_delta)
log_pkts	log(1 + Number)	log(1 + src_pkts + dst_pkts)	constant (single packets)

Table 11. Mapping of native labels to the seven harmonized classes.

Class	CIC-IoT2023	Edge-IIoTset	ToN-IoT
Benign	BenignTraffic	Normal	normal
DoS/DDoS	DDoS-, DoS-, Mirai-	DDoS_UDP, DDoS_ICMP, DDoS_TCP, DDoS_HTTP	ddos, dos
Recon	Recon-, VulnerabilityScan	Port_Scanning, Vulnerability_scanner, Fingerprinting	scanning
MITM/Spoofing	MITM-ArpSpoofing, DNS_Spoofing	MITM	mitm
BruteForce	DictionaryBruteForce	Password	password

Class	CIC-IoT2023	Edge-IIoTset	ToN-IoT
Web/Injection	SqlInjection, XSS, CommandInjection, Uploading_Attack, BrowserHijacking	SQL_injection, XSS, Uploading	injection, xss
Malware	Backdoor_Malware	Backdoor, Ransomware	backdoor, ransomware

Table 12. Cross-dataset results for every transfer direction (% , mean over three seeds, source-statistics normalization). ID = in-domain macro-F1 of the teacher; X = seven-class macro-F1 on the target dataset; Bin = binary attack-vs-benign F1 on the target.

Train to test	ID (T)	X Teacher	X S-CE	X S-LA	X KD	X MA-KD	Bin Teacher	Bin MA-KD
CIC-IoT2023 to Edge-IIoTset	51.4	11.1	11.4	10.6	7.5	4.9	60.1	61.8
CIC-IoT2023 to ToN-IoT	51.4	6.0	8.7	10.8	11.2	5.4	53.2	54.5
CIC-IoT2023+Edge-IIoTset to ToN-IoT	52.1	7.2	4.4	3.7	3.2	3.0	46.8	54.4
CIC-IoT2023+ToN-IoT to Edge-IIoTset	65.7	8.2	18.2	25.3	11.2	17.9	43.9	64.0
Edge-IIoTset to CIC-IoT2023	60.0	11.8	2.4	8.3	0.8	1.8	82.0	98.8
Edge-IIoTset to ToN-IoT	60.0	6.9	8.3	0.7	6.0	4.2	39.1	54.5
Edge-IIoTset+ToN-IoT to CIC-IoT2023	72.8	9.0	6.8	5.3	6.4	9.0	47.4	45.5
ToN-IoT to CIC-IoT2023	77.8	6.0	11.4	12.3	9.8	12.6	28.4	67.4
ToN-IoT to Edge-IIoTset	77.8	10.2	14.0	13.0	17.5	11.6	22.4	53.5

6. Discussion

6.1. Practical Guidance

Three recommendations follow from the results. First, evaluations of distilled IDS models should report per-class or at least macro-level metrics and a minority-recall figure, together with the same student trained without a teacher, so that capacity loss and distillation loss can be told apart. Our KD students keep accuracy above 89% while missing most ransomware, injection and man-in-the-middle flows, and a paper reporting accuracy alone would not reveal this. Second, if a distillation pipeline is used (for example, because a large model already exists and its training data cannot be reused), the distillation term must be made class-aware; class weighting and prior debiasing are simple to add and recover most of the loss. Third, standard imbalance-aware training of the small model should always be run as a baseline. In our experiments it was at least as good as the class-aware distilled model, and it needs neither a teacher nor stored teacher logits.

6.2. Limitations

Several limitations qualify these conclusions. The evidence across benchmarks rests on three datasets, so it shows consistency of direction and size but cannot establish significance across datasets. Partitions were random at the record level, and records from the same capture, session or device can appear on both sides of a split; the results apply to held-out records from the same collections. We used one teacher architecture, trained once per dataset; results with other teachers, or with students trained on less data where soft labels usually help more, may differ. The student is a single small MLP; larger students or other

architectures may benefit more from distillation. KD was run with one temperature and weight, and no method, including MA-KD, was tuned. The original BKD loss was compared analytically, not retrained, and the per-class predictions of S-WCE were not analysed. Class priors were taken from the sampled training split, which the 5000-record floor shifts toward the rarest classes. Latency was emulated on a cloud CPU with one thread, not measured on physical IoT hardware, and runtime memory was not measured. Only exact duplicates were removed, so near-duplicate flows may remain across splits. The minority threshold of 2% is a convention and places ten of fifteen Edge-IIoTset classes in the minority set. We did not evaluate robustness to adversarial or evasive traffic. The attribution analysis measures agreement with the teacher, not faithfulness, and the cross-dataset study confounds dataset shift with information lost in the harmonized schema.

7. Conclusions

Knowledge distillation, applied as usually described, can make lightweight IoT intrusion detectors forget rare attacks: across three datasets a distilled 5–9K-parameter student lost up to 65 points of minority-class recall relative to its teacher, and on two datasets distillation itself removed 14.9 and 54.3 points that the same student learns without a teacher. Class-weighted distillation from prior-debiased teacher logits removes this failure and improves minority recall over standard KD on all three datasets, with a 21 KB model that classifies a flow in about 14 μ s on one cloud CPU thread. The same experiments show that a class-weighted student trained without a teacher performs as well or better, so the value of distillation for very small IDS models remains unproven. Future work should test the limited-data regime, where teacher knowledge is expected to matter more, use grouped or temporal splits where capture identifiers exist, measure memory and latency on physical devices, and build cross-dataset benchmarks from raw traffic with a shared feature extractor.

Funding: This research received no external funding.

Data Availability Statement: CIC-IoT2023, Edge-IIoTset and ToN-IoT are publicly available from their original sources [6,7,24].

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

References

1. Hinton, G.; Vinyals, O.; Dean, J. Distilling the Knowledge in a Neural Network. *arXiv* 2015, arXiv:1503.02531.
2. Gou, J.; Yu, B.; Maybank, S.J.; Tao, D. Knowledge Distillation: A Survey. *Int. J. Comput. Vis.* 2021, 129, 1789–1819.
3. Wang, Z.; Zhou, R.; Yang, S.; He, D.; Chan, S. A Novel Lightweight IoT Intrusion Detection Model Based on Self-Knowledge Distillation. *IEEE Internet Things J.* 2025, 12, 16912–16930. <https://doi.org/10.1109/JIOT.2025.3533092>.
4. Wisanwanichthan, T.; Thammawichai, M. A Lightweight Intrusion Detection System for IoT and UAV Using Deep Neural Networks with Knowledge Distillation. *Computers* 2025, 14, 291. <https://doi.org/10.3390/computers14070291>.
5. Yang, S.; Zheng, X.; Xu, Z.; Wang, X. A Lightweight Approach for Network Intrusion Detection Based on Self-Knowledge Distillation. In Proceedings of the IEEE International Conference on Communications (ICC), Rome, Italy, 2023; arXiv:2307.10191.
6. 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. <https://doi.org/10.3390/s23135941>.
7. Moustafa, N. A New Distributed Architecture for Evaluating AI-Based Security Systems at the Edge: Network TON_IoT Datasets. *Sustain. Cities Soc.* 2021, 72, 102994. <https://doi.org/10.1016/j.scs.2021.102994>.
8. Hooker, S.; Courville, A.; Clark, G.; Dauphin, Y.; Frome, A. What Do Compressed Deep Neural Networks Forget? *arXiv* 2019, arXiv:1911.05248.
9. Gorishniy, Y.; Rubachev, I.; Khrulkov, V.; Babenko, A. Revisiting Deep Learning Models for Tabular Data. *Adv. Neural Inf. Process. Syst.* 2021, 34, 18932–18943.
10. Zhao, B.; Cui, Q.; Song, R.; Qiu, Y.; Liang, J. Decoupled Knowledge Distillation. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), New Orleans, LA, USA, 2022; pp. 11953–11962.
11. Menon, A.K.; Jayasumana, S.; Rawat, A.S.; Jain, H.; Veit, A.; Kumar, S. Long-Tail Learning via Logit Adjustment. In Proceedings of the International Conference on Learning Representations (ICLR), Virtual, 2021.
12. Zhang, S.; Chen, C.; Hu, X.; Peng, S. Balanced Knowledge Distillation for Long-Tailed Learning. *Neurocomputing* 2023, 527, 36–46. <https://doi.org/10.1016/j.neucom.2023.01.063>.
13. Wang, Z.; Li, Z.; He, D.; Chan, S. A Lightweight Approach for Network Intrusion Detection in Industrial Cyber-Physical Systems Based on Knowledge Distillation and Deep Metric Learning. *Expert Syst. Appl.* 2022, 206, 117671. <https://doi.org/10.1016/j.eswa.2022.117671>.
14. Wang, L.-H.; Dai, Q.; Du, T.; Chen, L.-F. Lightweight Intrusion Detection Model Based on CNN and Knowledge Distillation. *Appl. Soft Comput.* 2024, 165, 112118. <https://doi.org/10.1016/j.asoc.2024.112118>.
15. Xie, B.; Wang, Z.; Zeng, Z.; He, D.; Chan, S. DTKD-IDS: A Dual-Teacher Knowledge Distillation Intrusion Detection Model for the Industrial Internet of Things. *Ad Hoc Netw.* 2025, 174, 103869. <https://doi.org/10.1016/j.adhoc.2025.103869>.
16. AL-Nomasy, N.; Alamri, A.; Aljuhani, A.; Kumar, P. Transformer-Based Knowledge Distillation for Explainable Intrusion Detection System. *Comput. Secur.* 2025, 154, 104417. <https://doi.org/10.1016/j.cose.2025.104417>.
17. Yagiz, M.A.; Goktas, P. LENS-XAI: Redefining Lightweight and Explainable Network Security through Knowledge Distillation and Variational Autoencoders for Scalable Intrusion Detection in Cybersecurity. *arXiv* 2025, arXiv:2501.00790.
18. Cui, Y.; Jia, M.; Lin, T.-Y.; Song, Y.; Belongie, S. Class-Balanced Loss Based on Effective Number of Samples. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Long Beach, CA, USA, 2019; pp. 9268–9277.
19. Lin, T.-Y.; Goyal, P.; Girshick, R.; He, K.; Dollár, P. Focal Loss for Dense Object Detection. In Proceedings of the IEEE International Conference on Computer Vision (ICCV), Venice, Italy, 2017; pp. 2980–2988.
20. Ren, J.; Yu, C.; Sheng, S.; Ma, X.; Zhao, H.; Yi, S.; Li, H. Balanced Meta-Softmax for Long-Tailed Visual Recognition. *Adv. Neural Inf. Process. Syst.* 2020, 33, 4175–4186.
21. 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, Boston, MA, USA, 2022; pp. 3971–3988.
22. Engelen, G.; Rimmer, V.; Joosen, W. Troubleshooting an Intrusion Detection Dataset: The CICIDS2017 Case Study. In Proceedings of the IEEE Security and Privacy Workshops (SPW), San Francisco, CA, USA, 2021; pp. 7–12.
23. Sarhan, M.; Layeghy, S.; Portmann, M. Towards a Standard Feature Set for Network Intrusion Detection System Datasets. *Mob. Netw. Appl.* 2022, 27, 357–370. <https://doi.org/10.1007/s11036-021-01843-0>.

24. Ferrag, M.A.; Friha, O.; Hamouda, D.; Maglaras, L.; Janicke, H. Edge-IIoTset: A New Comprehensive Realistic Cyber Security Dataset of IoT and IIoT Applications for Centralized and Federated Learning. *IEEE Access* 2022, 10, 40281–40306. <https://doi.org/10.1109/ACCESS.2022.3165809>.
25. Loshchilov, I.; Hutter, F. Decoupled Weight Decay Regularization. In Proceedings of the International Conference on Learning Representations (ICLR), New Orleans, LA, USA, 2019.
26. Dietterich, T.G. Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. *Neural Comput.* 1998, 10, 1895–1923.
27. Efron, B.; Tibshirani, R.J. *An Introduction to the Bootstrap*; Chapman & Hall: New York, NY, USA, 1993.
28. Demšar, J. Statistical Comparisons of Classifiers over Multiple Data Sets. *J. Mach. Learn. Res.* 2006, 7, 1–30.
29. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and Training of Neural Networks for Efficient Integer-Arithmetic-Only Inference. In Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), Salt Lake City, UT, USA, 2018; pp. 2704–2713.
30. ONNX Runtime. Available online: <https://onnxruntime.ai> (accessed on 8 October 2026).
31. Alhijawi, B., Kilani, Y., & Alsarhan, A. (2020). Improving recommendation quality and performance of genetic-based recommender system. *International Journal of Advanced Intelligence Paradigms*, 15(1), 77-88.
32. Sundararajan, M.; Taly, A.; Yan, Q. Axiomatic Attribution for Deep Networks. In Proceedings of the 34th International Conference on Machine Learning (ICML), Sydney, Australia, 2017; pp. 3319–3328.