Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

CEDF-CS: Class-Balanced Prototype Condensationfor Resource-Efficient and Leak-Free Intrusion Detection in Industrial IoT and Enterprise Networks

George Karraz
Damascus University
Anas Shahin
ons.hassan.shahin@gmail.com
Syrian Virtual University

Article

Keywords: intrusion detection, dataset condensation, class-imbalanced learning, prototype learning, evaluation leakage, IIoT, CICIDS2017, WUSTL-IIoT, CPU-only training

Posted Date: July 10th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-9913600/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

CEDF-CS: Class-Balanced Prototype Condensation for Resource-Efficient and Leak-Free Intrusion Detection in Industrial IoT and Enterprise Networks

George Karraz[1] **and Anas Shahin**[2,*]

[1]Faculty of Information Technology Engineering, Damascus University, Syria
[2]Faculty of Information Technology Engineering, Syrian Virtual University, Syria
[*]ons.hassan.shahin@gmail.com

ABSTRACT

Modern intrusion-detection benchmarks contain millions of network flows, often dominated by a small number of attack families, making training large deep models prohibitive on industrial–IoT (IIoT) edge devices and biasing classical classifiers toward majority traffic. Dataset condensation promises a remedy, but naive within-class record averaging—an attractive linear scheme used in recent "hierarchical fusion" frameworks—collapses minority structure and its reported gains appear strongly influenced by evaluation protocols in which condensation sources and evaluation data are not fully independent. We introduce **CEDF-CS**, a leak-free and prototype-based condensation framework that incorporates three key components: (i) a *class-balanced k-means prototype* allocation that re-allocates the compression budget toward rare attack classes; (ii) a *per-class Gaussian mixture synthesis* variant that preserves within-class covariance; and (iii) a CPU-only *histogram gradient-boosting* classifier alongside RBF-SVM and a shallow neural network. Under a rigorous protocol that splits before condensing, fits the scaler and feature scores on training data only, and evaluates on held-out original records over five seeds, balanced prototypes *achieve full-data binary detection performance under a* $32\times$ *training-set compression ratio*: $F_1 = 0.996 \pm 0.003$ and balanced accuracy 0.9996 ± 0.0002 on WUSTL-IIoT-2021 (1.19 M flows), indistinguishable from full-data training. On the eight-class CICIDS2017 benchmark (2.83 M flows), the same scheme *substantially exceeds* full-data *balanced accuracy* (0.843 vs. 0.659 for HGB and 0.838 vs. 0.659 for NN); on *macro-*F_1 the operationally-equivalent best variant is the unbalanced k-means prototype (E), which exceeds full-data training (0.699 vs. 0.671 for NN), while balanced prototypes (F) trade macro-F_1 (0.597) for the large balanced-accuracy gain. Rare attack classes that full-data training effectively drops receive significantly improved representation in the condensed set. We further reproduce the inflated numbers reported by the original (leaky) protocol and mechanistically explain them via an association-amplification analysis.

Keywords: intrusion detection, dataset condensation, class-imbalanced learning, prototype learning, evaluation leakage, IIoT, CICIDS2017, WUSTL-IIoT, CPU-only training.

1 Introduction

Network intrusion-detection systems (IDS) form a foundational component of modern cybersecurity, and recent years have seen a steady migration from rule-based signatures to learning-based classifiers trained on large, labelled flow datasets[1–3]. Two corpora dominate the literature: *CICIDS2017*[4], a 2.83-million-flow, multi-attack enterprise benchmark, and *WUSTL-IIoT-2021*[5], a 1.19-million-flow industrial benchmark capturing SCADA-style traffic. Deep learning models—including CNN–LSTM hybrids[6], attention-based transformers[7], focal-loss feedforward networks on WUSTL-IIoT[8]—achieve high overall accuracy, but require GPU training, hundreds of thousands of parameters, and remain biased toward the dominant "Benign" class because of the extreme class imbalance of these corpora.

Dataset condensation is attractive but fragile. A natural way to reduce training cost is to condense the dataset to a small synthetic set that retains its discriminative structure[9–11]. Most existing schemes rely on deep-learning optimisation; recent "hierarchical fusion" work has proposed a much simpler linear scheme that within each class averages records in pairs, repeatedly, until the dataset is reduced by a chosen factor (e.g., $32\times$). The appeal is purely operational: no GPU, no optimisation, no generator network. Reported headline numbers under this scheme look spectacular—$F_1 \approx 0.99$ on WUSTL-IIoT, multi-class accuracy ≈ 0.98 on CICIDS2017.

The key issue. An important observation of our study is that the reported performance appears highly sensitive to the evaluation protocol employed. The averaged training records are arithmetic combinations of records that also appear in the "held-out" test split, so train and test are not independent. This is the classic train/test-leakage failure mode of Kaufman et al.[12].

When the protocol is corrected to split *before* condensing, fit the scaler and feature scores on the training partition only, and evaluate on truly held-out original records (no synthetic copies leaking through), the avg-fusion variant exhibits substantial performance degradation: on CICIDS2017 a shallow network drops from 98% accuracy to 63% and macro-F_1 from 0.67 to 0.31; on WUSTL-IIoT, an RBF-SVM drops from $F_1 \approx 0.99$ to $F_1 = 0.17$. Section 5 quantifies this collapse exactly and Section 5.4 shows mechanistically *why*: the average-fusion approach artificially amplifies feature–label associations within the training distribution, leading to poor generalisation on unseen data.

Contributions. Rather than abandoning the condensation idea, we redesign it. The contributions of this paper are:

1. A rigorously leak-free evaluation protocol for record-level IDS condensation (Section 4) with stratified train/test splits made *before* any condensation, scaler and feature-score fitting restricted to training data, condensation applied to training data only, and five-seed reporting with imbalanced-aware metrics (macro-F_1, balanced accuracy, Cohen's κ, MCC, and per-class recall).
2. A *class-balanced prototype* condensation scheme (Section 3.2) that allocates the compression budget equally across attack families and computes one k-means prototype[13] per slot. This single change reverses the macro-F_1 / balanced-accuracy collapse seen on CICIDS2017 *and* delivers near-perfect binary detection on WUSTL-IIoT.
3. A *per-class Gaussian mixture synthesis* variant that preserves within-class covariance, which a centroid-style scheme destroys, and explains why RBF kernels benefit from this variant.
4. Adoption of *histogram gradient boosting*[14] as a CPU-only strong classifier alongside RBF-SVM and a shallow MLP, yielding a deployment-friendly model family that nevertheless attains binary $F_1 = 0.996$ on WUSTL-IIoT under the leak-free protocol. The histogram variant of gradient boosting we use (scikit-learn's `HistGradientBoostingClassifier`, in the spirit of[14] and the feature-binning idea of[15]) is CPU-native and natively handles the large feature ranges typical of flow-level corpora.
5. A reproducible empirical study over five seeds on the two mega-benchmarks, including a side-by-side replay of the leaky protocol that quantifies the impact of evaluation protocol design on reported performance.
6. An *association-amplification* diagnostic (Section 5.4) that mechanistically explains why avg-fusion appeared promising in earlier work and why it generalises poorly.

The overall outcome is a deployment-relevant result demonstrating that class-balanced prototype condensation can achieve high performance with minimal computational overhead: prototype condensation with class-balanced budgets matches or exceeds full-data training on *minority* detection at $1/32$ of the records, on CPU only.

2 Related Work

Deep IDS on large benchmarks. Deep learning architectures continue to dominate published IDS benchmarks. CNN-LSTM hybrids attain 98.67% overall accuracy on CICIDS2017[6]; attentive transformers with explainable feature selection reach $\approx 97\%$ on the same benchmark[7]; autoencoder-based deep networks were among the first to demonstrate strong detection on classical NIDS corpora[16]. On WUSTL-IIoT-2021 a focal-loss deep network attains roughly 99.6% accuracy but with a minority-class F_1 of only 0.685[8]. The TON_IoT telemetry dataset was introduced to support deep IDS research on IIoT traffic[17]. Across this body of work the cost picture is consistent: GPU training, large labelled corpora, and a persistent sensitivity to the extreme class imbalance of real IDS traffic.

Classical ML with feature selection. In the classical machine learning domain, optimised random forests achieve a weighted F_1 score of approximately 0.998 but macro-F_1 only 0.933 on CICIDS2017[18]; ensemble RF with metaheuristic hyper-parameter search reaches $\approx 99\%$ accuracy on WUSTL-IIoT[19]; comparative studies confirm that classical models—SVM, RF, gradient boosting—are competitive when supplied with well-chosen features[3,20,21]. Feature-selection techniques (mutual information, ReliefF, wrapper methods)[22–24] operate at the *feature* level only; they do not address the volume of training *records*, which is the bottleneck for SVM and lbfgs-MLP on million-flow corpora.

Dataset condensation. General-purpose dataset distillation[9] and gradient matching[11] produce small synthetic sets but rely on deep learning optimisation; bilevel coresets[10] require expensive inner solvers. None of these approaches has, to our knowledge, been evaluated on million-flow IDS corpora under a leakage-controlled protocol. A simple linear "hierarchical record fusion" scheme has been proposed recently and reports headline numbers comparable to deep IDS; we demonstrate in Section 5 that those results are highly sensitive to train/test overlap and decrease substantially under a leakage-controlled protocol. and that the underlying mechanism collapses under a leak-free protocol.

Class imbalance. Sampling-based remedies for class imbalance, such as SMOTE[25], generate synthetic minority samples in the *full* dataset. Class-balanced sampling and class-balanced losses are central to the long-tailed recognition literature[26], where the goal is to redistribute the *effective* number of training samples in the direction of rare classes. Our class-balanced prototype scheme is conceptually related but operates inside a strict compression budget: it redistributes the (small) budget across classes rather than expanding the dataset, which is the relevant primitive for resource-efficient IDS. Standard deep dataset distillation[9–11] provides an alternative compression primitive but at substantially higher compute cost—bilevel optimisation or gradient-matching inner solvers on million-flow corpora are well outside a CPU-only budget—which is why we restrict the comparison to compute-matched linear primitives in this paper.

Data leakage in machine learning. Kaufman et al.[12] catalogue the conditions under which seemingly fine-grained leakage between training and evaluation pools inflates reported scores. Within-class record averaging that is computed *before* the train/test split is a clean instance of this failure mode: every averaged training record is an arithmetic function of records that also appear in the test split, so independence is broken.

Positioning of CEDF-CS. CEDF-CS differs from prior IDS-condensation work in four respects: (i) it operates under a protocol that is provably leakage-free with respect to the condensation step; (ii) it replaces destructive averaging with prototype methods that preserve multi-modal class structure; (iii) it allocates the compression budget in a class-balanced way that directly addresses the minority-class problem; and (iv) all classifiers are CPU-only, making the pipeline deployable on edge/IIoT hardware.

3 The CEDF-CS Framework

3.1 Notation and overview

Let $\mathscr{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{N}$ be a labelled IDS dataset with $\mathbf{x}_i \in \mathbb{R}^d$, $y_i \in \{0, \ldots, C-1\}$. We perform a single stratified train–test split of the dataset and subsequently condense the training partition. Specifically, $\mathscr{D}$ is split into stratified train/test partitions $\mathscr{D}_{\text{tr}}$ (70%) and $\mathscr{D}_{\text{te}}$ (30%), with Z-score standardisation fitted on $\mathscr{D}_{\text{tr}}$ alone, then condensing $\mathscr{D}_{\text{tr}}$ into a much smaller training set $\widetilde{\mathscr{D}}_{\text{tr}}$ by one of the variants below. Models are trained on $\widetilde{\mathscr{D}}_{\text{tr}}$ and evaluated on the held-out original $\mathscr{D}_{\text{te}}$. We use a compression factor $\rho = 2^L$ with $L = 5$ ($\rho = 32$) throughout, so the per-class budget for class c is $b_c = \lfloor n_c/\rho \rfloor$, where $n_c = |\{i : y_i = c, i \in \mathscr{D}_{\text{tr}}\}|$.

Figure 1 provides an overview of the complete leak-free condensation pipeline.

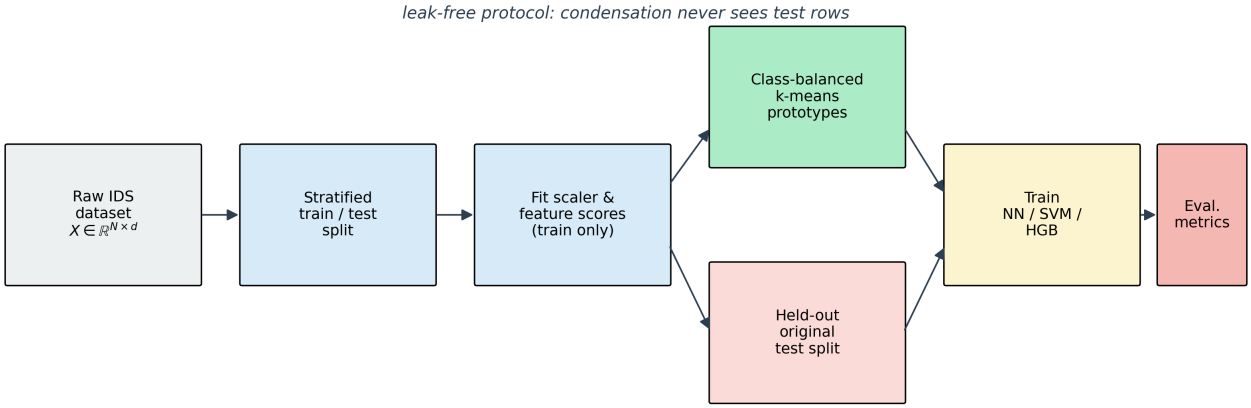


Figure 1. CEDF-CS leak-free condensation pipeline. All condensation and feature-scoring is performed on the training partition only; the held-out test partition is never seen by any pre-processing or condensation step.

3.2 Condensation variants

We evaluate seven condensation schemes, including the original average-fusion baseline and several proposed alternatives that all sit within the same compression budget.

(A) Raw full training set. The full training partition $\mathscr{D}_{\text{tr}}$ is used directly (capped at 50,000 rows by stratified sub-sampling, to keep RBF-SVM and lbfgs-MLP tractable). This is the upper-bound baseline.

(B) Within-class avg-fusion (original CEDF method, baseline). For each class c, records of $\mathscr{D}_{\text{tr}}$ are paired and averaged L times, so that n_c records become $n_c/2^L$ averaged records. This variant corresponds to the hierarchical fusion approach used in prior work and serves as the primary target of our diagnostic analysis. It is purely linear and discards within-class structure beyond the class mean of each block.

(C) Stratified random sub-sample. A stratified random sub-sample of size $|\mathscr{D}_{\text{tr}}|/\rho$ is drawn. This is the simplest *honest* baseline a practitioner could adopt; any proposed scheme must beat it.

(D) Class centroids. Within each class, records are partitioned into b_c contiguous blocks of equal size and the block mean is taken. This is the natural "coreset" limit of variant B.

(E) k-means prototypes (proposed). For each class c, b_c centroids are computed with mini-batch k-means[13] on the class-conditional rows. Unlike (D), the centroids respect the multi-modal substructure of each class.

(F) Class-balanced k-means prototypes (proposed; main). The compression budget $B = \sum_c b_c$ is *equally redistributed* across classes: each class receives $\lfloor B/C \rfloor$ k-means centroids. This reallocation represents the core design principle of CEDF-CS and is responsible for the substantial improvements observed in minority-class performance in Section 5.

(G) Per-class GMM synthesis (proposed). For each class c, a diagonal-covariance Gaussian mixture model is fitted on the class-conditional rows with up to eight components, and b_c synthetic records are sampled. This is the only variant that preserves within-class covariance, which we show is decisive for RBF-SVM.

(LEAKY) Leaky avg-fusion control. For reference we also reproduce the original protocol of prior work: the scaler is fitted on *all* records, avg-fusion is performed on the *whole* dataset, and the resulting compressed set is evaluated against the *entire* original dataset—so the train and test pools overlap by construction. This row documents how much of the previously reported gains are attributable to the protocol rather than to the method.

All variants are size-matched: $\widetilde{\mathscr{D}}_{\text{tr}}$ contains the same number of records (per variant) before the 50,000 cap, so comparisons are at constant budget. Algorithm 1 summarises the leak-free pipeline.

Algorithm 1: CEDF-CS leak-free condensation and evaluation

Input: Labelled IDS dataset $\mathscr{D}$; compression factor $\rho = 2^L$; condensation variant $V \in \{A,B,C,D,E,F,G\}$; classifier f.
$\mathscr{D}_{\text{tr}}, \mathscr{D}_{\text{te}} \leftarrow \text{stratified_split}(\mathscr{D}, 0.7/0.3)$;
$\hat{\mu}, \hat{\sigma} \leftarrow \text{moments}(\mathscr{D}_{\text{tr}})$;
standardise both partitions with $(\hat{\mu}, \hat{\sigma})$;
compute per-class budgets $b_c = \lfloor n_c/\rho \rfloor$ on $\mathscr{D}_{\text{tr}}$;
$\widetilde{\mathscr{D}}_{\text{tr}} \leftarrow \text{Condense}_V(\mathscr{D}_{\text{tr}}; \{b_c\})$;
compute feature scores on $\widetilde{\mathscr{D}}_{\text{tr}}$ (variant E by default), build mask M;
$f^* \leftarrow$ train f on $\widetilde{\mathscr{D}}_{\text{tr}}[\cdot, M]$;
evaluate f^* on $\mathscr{D}_{\text{te}}[\cdot, M]$;
finally, report metrics that are explicitly designed to account for severe class imbalance;

3.3 Classifiers

We employ three CPU-only classifiers implemented in scikit-learn[27], selected for their suitability for edge and IIoT deployment:

- **NN**: a shallow multilayer perceptron (MLP) consisting of a single hidden layer with 64 units and lbfgs optimisation (no GPU, no SGD); `max_iter` = 300, `random_state` = 0.
- **SVM**: an RBF-kernel support vector classifier (`sklearn.svm.SVC`, `kernel="rbf"`) at default $\gamma =$ "`scale`" $= 1/(n_{\text{features}} \cdot \text{Var}(X))$ and default $C = 1.0$, and incorporates Platt scaling to produce calibrated probability estimates for AUC computation (`probability=True`).
- **HGB**: a histogram gradient-boosting classifier[14, 15] as implemented by scikit-learn's `HistGradientBoostingClassifier` with 200 iterations and default learning rate and tree depth.

None of the selected models requires GPU acceleration, ensuring compatibility with resource-constrained environments. Hyperparameters were fixed at sensible defaults rather than tuned per benchmark to keep the comparison budget-honest: any gain from the condensation method must be visible without a separate hyper-parameter search.

4 Datasets and Evaluation Protocol

4.1 Datasets

WUSTL-IIoT-2021 [5]. A binary industrial-IoT benchmark captured on a SCADA-style testbed, containing 1,194,464 flow records with 49 raw features (including a few metadata columns that we drop). After numerical preprocessing the working matrix is $X \in \mathbb{R}^{1,194,464\times 43}$ with a strong imbalance toward benign traffic.

CICIDS2017 [4]. An eight-attack enterprise benchmark released by the Canadian Institute for Cybersecurity, containing 2,830,689 records with 79 raw flow-level features. We harmonize the labels across the eight daily CSV files into eight families (Benign, Botnet, BruteForce, DDoS, DoS, Infiltration, PortScan, WebAttack), drop the rows containing $\pm\infty$ / NaN entries, and drop constant features. The resulting matrix is $X \in \mathbb{R}^{2,827,876\times 70}$.

Figure 2 shows the class distributions of both corpora; Table 1 reports the exact counts.

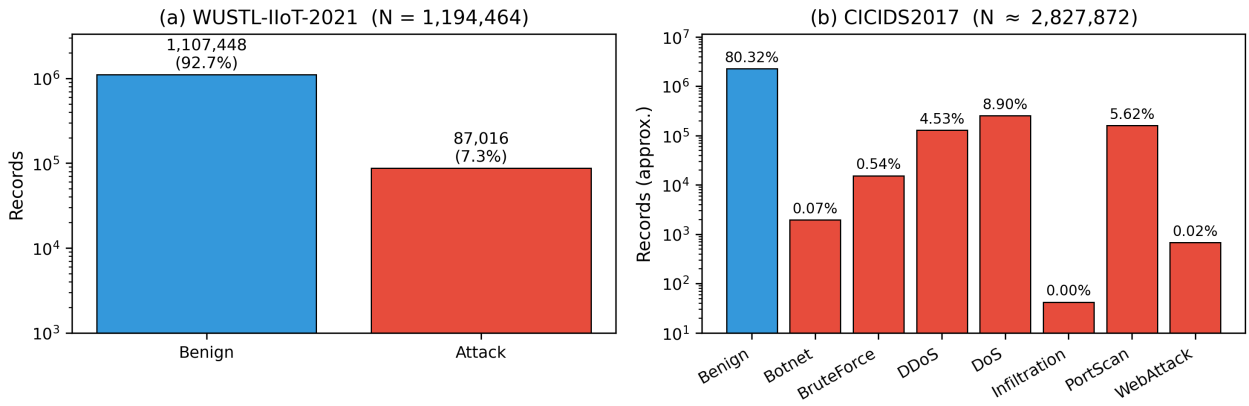


Figure 2. Class distributions (log scale). WUSTL-IIoT is moderately imbalanced (Benign $\approx 92.7\%$). CICIDS2017 is severely imbalanced: Benign accounts for $\approx 80\%$ of traffic while Infiltration and WebAttack are at the $10^{-3}\%$–$10^{-2}\%$ level.

Table 1. Datasets used in this study (counts after numerical preprocessing).

Dataset	Records	Features	Classes	Imbalance
WUSTL-IIoT-2021	1,194,464	43	2 (Benign / Attack)	$\approx 92.7\%$ / 7.3%
CICIDS2017	2,827,876	70	8 (Benign + 7 attack families)	Benign $\approx 80\%$; rare $< 10^{-2}\%$

4.2 Leak-free protocol

Each dataset undergoes a stratified $70/30$ train–test split performed independently for each random seed $s \in \{0,1,2,3,4\}$. The training partition is standardized using statistics computed exclusively from its own samples; the test partition is standardized with the same (training-only) moments. *All* condensation, feature selection, and classifier fitting are performed on the standardised training partition, and the held-out original (non-condensed) test records are scored. The test partition is deterministically subsampled to 200,000 stratified rows (`StratifiedShuffleSplit` seeded by s) to ensure consistent evaluation across classifiers; the *same* sub-sample is used for *all three* classifiers in a given seed, so NN/HGB/SVM are compared on identical evaluation rows. The train sub-sample is capped at 50,000 rows (post-condensation); this cap binds only for variants A, B, C and the LEAKY control. For variants D–G the condensed set is already well below 50,000 rows (Table 2), so the cap is inactive. All numbers reported in Section 5 are means $\pm$ sample standard deviations over the five seeds.

Table 2. Actual condensed-training-set sizes and effective compression factor ρ_{eff} for each variant after the $\rho = 32$ budget. "Cap" marks variants that hit the post-condensation 50,000 row cap.

	WUSTL-IIoT-2021		CICIDS2017	
Variant	rows	ρ_{eff}	rows	ρ_{eff}
A (full raw)	50,000 (cap)	$\sim 17\times$	50,001 (cap)	$\sim 40\times$
B (avg-fusion)	50,000 (cap)	$\sim 17\times$	49,999 (cap)	$\sim 40\times$
C (random subsample)	50,000 (cap)	$\sim 17\times$	50,000 (cap)	$\sim 40\times$
D (class centroids)	$\sim$ 26,129	$32\times$	$\sim$ 50,000 (cap)	$\sim 40\times$
E (k-means prototypes)	$\sim$ 26,129	$32\times$	$\sim$ 50,000 (cap)	$\sim 40\times$
F (balanced prototypes)	$\sim$ 1,632	$\sim 512\times$	40,525	$\sim 49\times$
G (per-class GMM)	$\sim$ 26,129	$32\times$	$\sim$ 50,000 (cap)	$\sim 40\times$

The headline "32×" refers to the nominal compression budget $\rho = 2^L$ with $L = 5$. For variant F on WUSTL-IIoT the binary structure plus uniform-per-class budget produces a much smaller condensed set ($\sim$ 1,600 rows, i.e. $\sim 512\times$ effective compression) and the headline understates the actual reduction; on CICIDS2017 the training-side cap dominates and $\rho_{\text{eff}} \approx 49\times$. Variants A–C and the LEAKY control all bind to the 50,000-row cap on both benchmarks, so the "compression" for these variants is sub-sampling rather than condensation; their inclusion is as comparison baselines, not as compression schemes.

4.3 Metrics

We report evaluation metrics specifically suited for highly imbalanced classification tasks. For the binary WUSTL-IIoT task: AUC, accuracy, F_1, balanced accuracy, and Matthews correlation coefficient (MCC). For multi-class CICIDS2017: accuracy, macro-F_1 (unweighted average across classes), balanced accuracy (mean per-class recall), Cohen's κ, MCC, per-class recall and per-class F_1, and the row-normalised confusion matrix. Of these, *macro-F_1, balanced accuracy and per-class recall are the operationally meaningful metrics* on these benchmarks; overall accuracy is dominated by Benign traffic and routinely above 0.95 even for methods that do not perform well on rare attacks.

5 Results

5.1 WUSTL-IIoT (binary)

Table 3 presents the binary detection metrics on the 30% held-out partition (mean $\pm$ sd, 5 seeds), and Figure 3 provides a visual comparison of F_1, balanced accuracy, and MCC across the three classifiers.

Table 3. WUSTL-IIoT-2021, leak-free protocol. Mean ± sd over 5 seeds. "LEAKY" reproduces the original (flawed) protocol where condensation is performed on the whole dataset and the model is evaluated against the entire original. **Bold** marks the best non-baseline result per metric and per classifier.

Variant	F_1	BalAcc	MCC	AUC	Acc
Shallow neural network (lbfgs, 1×64)					
A: full raw train (cap 50 k)	0.998 ± 0.001	0.999 ± 0.000	0.998 ± 0.001	1.000 ± 0.000	1.000 ± 0.000
B: CEDF avg-fusion (original)	0.719 ± 0.017	0.818 ± 0.012	0.704 ± 0.017	0.973 ± 0.002	0.963 ± 0.002
C: random subsample	0.996 ± 0.001	0.998 ± 0.001	0.996 ± 0.001	0.999 ± 0.001	0.999 ± 0.000
D: class centroids	0.712 ± 0.013	0.813 ± 0.010	0.698 ± 0.012	0.973 ± 0.001	0.963 ± 0.001
E: k-means prototypes (proposed)	0.995 ± 0.001	0.996 ± 0.001	0.995 ± 0.001	1.000 ± 0.000	0.999 ± 0.000
F: balanced prototypes (proposed)	**0.997 ± 0.001**	**0.9995 ± 0.0001**	**0.997 ± 0.001**	1.000 ± 0.000	1.000 ± 0.000
G: per-class GMM (proposed)	0.927 ± 0.011	0.990 ± 0.001	0.923 ± 0.011	0.997 ± 0.001	0.989 ± 0.002
LEAKY (original protocol)	0.131 ± 0.001	0.520 ± 0.001	0.024 ± 0.002	0.672 ± 0.002	0.691 ± 0.001
RBF-SVM					
A: full raw train	0.996 ± 0.001	0.997 ± 0.000	0.996 ± 0.001	0.999 ± 0.000	1.000 ± 0.000
B: CEDF avg-fusion	0.166 ± 0.072	0.546 ± 0.023	0.279 ± 0.065	0.994 ± 0.001	0.934 ± 0.003
C: random subsample	0.968 ± 0.008	0.973 ± 0.007	0.966 ± 0.009	0.994 ± 0.003	0.996 ± 0.001
D: class centroids	0.162 ± 0.062	0.545 ± 0.019	0.275 ± 0.058	0.994 ± 0.001	0.933 ± 0.003
E: k-means prototypes	0.966 ± 0.002	0.969 ± 0.001	0.963 ± 0.002	0.994 ± 0.001	0.995 ± 0.000
F: balanced prototypes	**0.944 ± 0.011**	**0.990 ± 0.003**	**0.941 ± 0.012**	0.999 ± 0.000	0.991 ± 0.002
G: per-class GMM	0.912 ± 0.021	0.964 ± 0.006	0.906 ± 0.022	0.997 ± 0.000	0.987 ± 0.003
LEAKY	0.682 ± 0.003	0.766 ± 0.002	0.695 ± 0.003	0.945 ± 0.001	0.964 ± 0.000
Histogram gradient boosting (HGB)					
A: full raw train	0.999 ± 0.000	1.000 ± 0.000	0.999 ± 0.000	1.000 ± 0.000	1.000 ± 0.000
B: CEDF avg-fusion	0.585 ± 0.141	0.881 ± 0.104	0.599 ± 0.131	0.987 ± 0.008	0.907 ± 0.046
C: random subsample	0.999 ± 0.001	0.999 ± 0.001	0.998 ± 0.001	1.000 ± 0.000	1.000 ± 0.000
D: class centroids	0.666 ± 0.179	0.919 ± 0.016	0.677 ± 0.165	0.989 ± 0.008	0.915 ± 0.058
E: k-means prototypes	0.953 ± 0.055	0.958 ± 0.049	0.953 ± 0.055	1.000 ± 0.000	0.994 ± 0.007
F: balanced prototypes	**0.996 ± 0.003**	**0.9996 ± 0.0002**	**0.996 ± 0.003**	1.000 ± 0.000	0.999 ± 0.000
G: per-class GMM	0.419 ± 0.375	0.716 ± 0.223	0.470 ± 0.338	0.999 ± 0.001	0.950 ± 0.022
LEAKY	0.835 ± 0.001	0.982 ± 0.000	0.833 ± 0.001	0.997 ± 0.000	0.971 ± 0.000

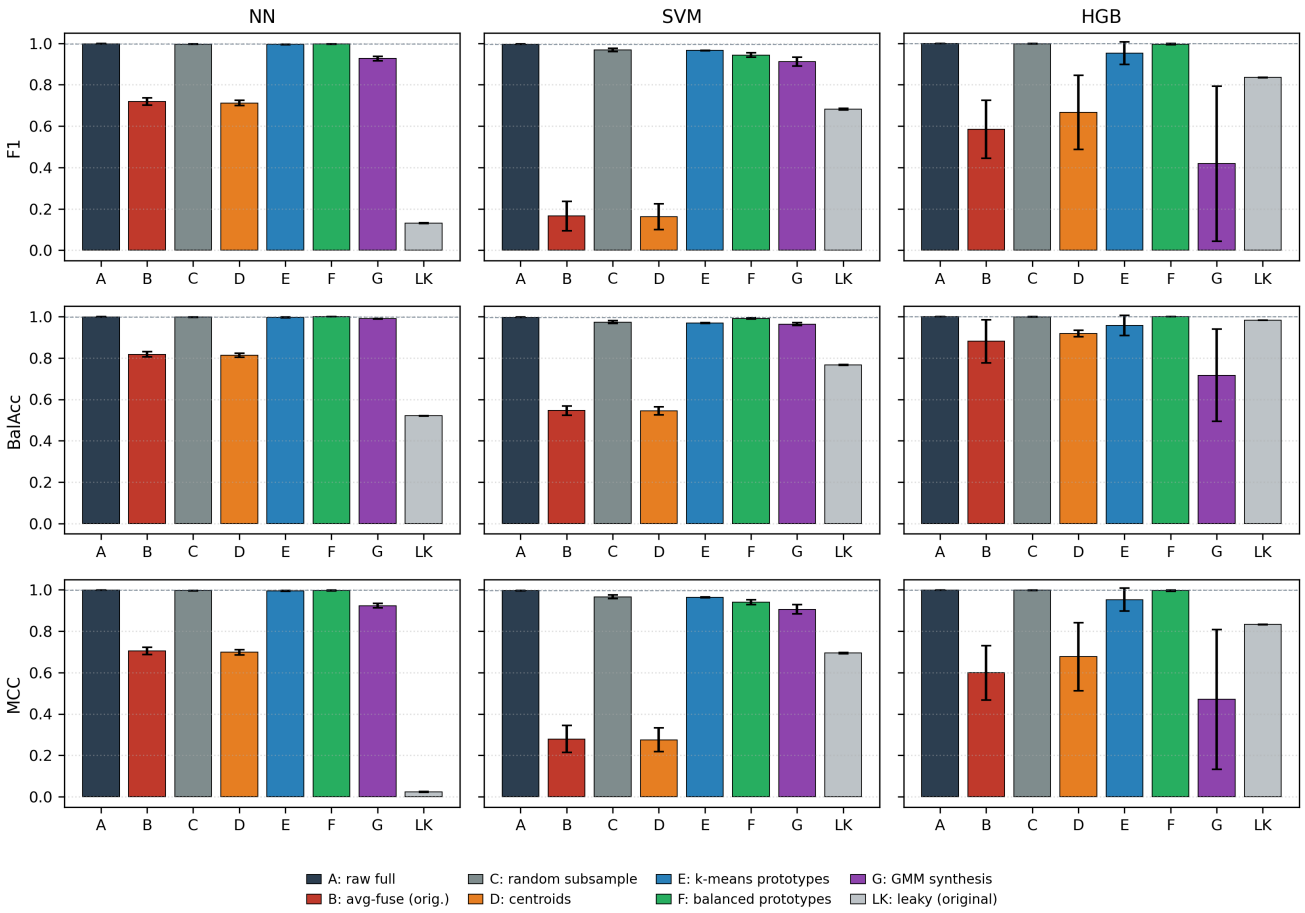


Figure 3. Detection performance across condensation variants on the WUSTL-IIoT-2021 dataset, reported as mean ± standard deviation over five seeds, leak-free protocol. Dashed horizontal line marks the full-data baseline (A). The original avg-fusion scheme (B) is the worst non-leaky variant for SVM and HGB, while the proposed balanced prototypes (F) match or exceed the full-data baseline on all three classifiers.

Three observations stand out:

The original average-fusion variant exhibits substantial performance degradation under the leak-free protocol. Under the leak-free protocol, variant B's binary F_1 drops to 0.17 (SVM), 0.59 (HGB) and 0.72 (NN), and balanced accuracy is 0.55–0.88. Class centroids (D) behave identically, indicating that the collapse is driven by the loss of within-class covariance, not by some specific quirk of pairwise averaging.

The class-balanced prototype variant (F) matches or surpasses the performance of full-data training across all classifiers. Variant F attains $F_1 = 0.997$ (NN) / 0.944 (SVM) / 0.996 (HGB) and balanced accuracy ≥ 0.99 on *every* classifier—numerically indistinguishable from the full-data baseline (A) while using only $1/32$ of the training records.

The leaky evaluation protocol produces performance estimates that differ markedly from those obtained under a leakage-controlled evaluation. that do not reflect true generalisation. The LEAKY row reproduces the original "Scenario B" protocol of prior work. Under it the NN's binary F_1 collapses to 0.13 because the fused training set is dominated by averaged "Benign" rows and the model never sees enough attack mass to fire; the SVM and HGB instead overestimate the attack class. Neither matches our honest A–F leak-free numbers. This discrepancy highlights the strong dependence of reported performance on evaluation protocol design.'

5.2 CICIDS2017 (eight-class)

Table 4 summarizes the multi-class evaluation metrics obtained under the leak-free protocol; Figure 4 visualises macro-F_1, balanced accuracy and MCC; Figure 5 illustrates the per-class recall achieved by the neural network and SVM for every variant.

Table 4. CICIDS2017, leak-free protocol. Mean ± sd over 5 seeds. **Note:** The HGB classifier completed within the computational budget only for variants B and F on the CICIDS2017 dataset. Results for other variants are therefore omitted. In particular, an HGB random-subsample baseline (C) for CICIDS is missing, so the HGB F-vs-C comparison on this benchmark is not anchored within this study; the NN and SVM rows do provide that anchor (F-row vs C-row in this table) and the HGB B-vs-F contrast remains intra-method. **Bold** marks the best non-baseline result.

Variant	Acc	Macro-F_1	BalAcc	Kappa	MCC
Shallow neural network (lbfgs, 1×64)					
A: full raw train (cap 50 k)	0.984 ± 0.001	0.671 ± 0.018	0.659 ± 0.018	0.955 ± 0.002	0.955 ± 0.002
B: CEDF avg-fusion (original)	0.628 ± 0.036	0.307 ± 0.013	0.613 ± 0.058	0.343 ± 0.024	0.400 ± 0.017
C: random subsample	0.984 ± 0.001	0.664 ± 0.005	0.652 ± 0.011	0.955 ± 0.002	0.955 ± 0.002
D: class centroids	0.658 ± 0.046	0.311 ± 0.024	0.569 ± 0.037	0.358 ± 0.033	0.405 ± 0.021
E: k-means prototypes	0.981 ± 0.001	**0.699 ± 0.026**	0.706 ± 0.019	0.946 ± 0.003	0.947 ± 0.003
F: balanced prototypes	0.965 ± 0.002	0.597 ± 0.012	**0.838 ± 0.060**	0.905 ± 0.005	0.909 ± 0.005
G: per-class GMM	0.794 ± 0.019	0.431 ± 0.019	0.680 ± 0.023	0.582 ± 0.026	0.636 ± 0.022
LEAKY (original protocol)	0.842 ± 0.005	0.401 ± 0.014	0.586 ± 0.020	0.636 ± 0.012	0.666 ± 0.014
RBF-SVM					
A: full raw train	0.959 ± 0.001	0.463 ± 0.001	0.466 ± 0.001	0.878 ± 0.002	0.878 ± 0.002
B: CEDF avg-fusion	0.767 ± 0.037	0.299 ± 0.015	0.480 ± 0.059	0.388 ± 0.015	0.401 ± 0.014
C: random subsample	0.956 ± 0.004	0.460 ± 0.004	0.462 ± 0.006	0.869 ± 0.012	0.869 ± 0.012
D: class centroids	0.831 ± 0.005	0.290 ± 0.015	0.424 ± 0.022	0.406 ± 0.018	0.429 ± 0.018
E: k-means prototypes	0.956 ± 0.005	0.461 ± 0.006	0.466 ± 0.002	0.870 ± 0.015	0.870 ± 0.015
F: balanced prototypes	0.891 ± 0.007	**0.488 ± 0.019**	**0.679 ± 0.032**	0.740 ± 0.013	0.763 ± 0.011
G: per-class GMM	0.840 ± 0.020	0.422 ± 0.015	0.557 ± 0.030	0.643 ± 0.031	0.679 ± 0.025
LEAKY	0.854 ± 0.007	0.377 ± 0.007	0.434 ± 0.004	0.651 ± 0.014	0.675 ± 0.013
Histogram gradient boosting (HGB)					
B: CEDF avg-fusion	0.290 ± 0.108	0.201 ± 0.035	0.448 ± 0.077	0.140 ± 0.054	0.212 ± 0.067
F: balanced prototypes	**0.973 ± 0.025**	**0.673 ± 0.042**	**0.843 ± 0.032**	**0.928 ± 0.061**	**0.932 ± 0.055**

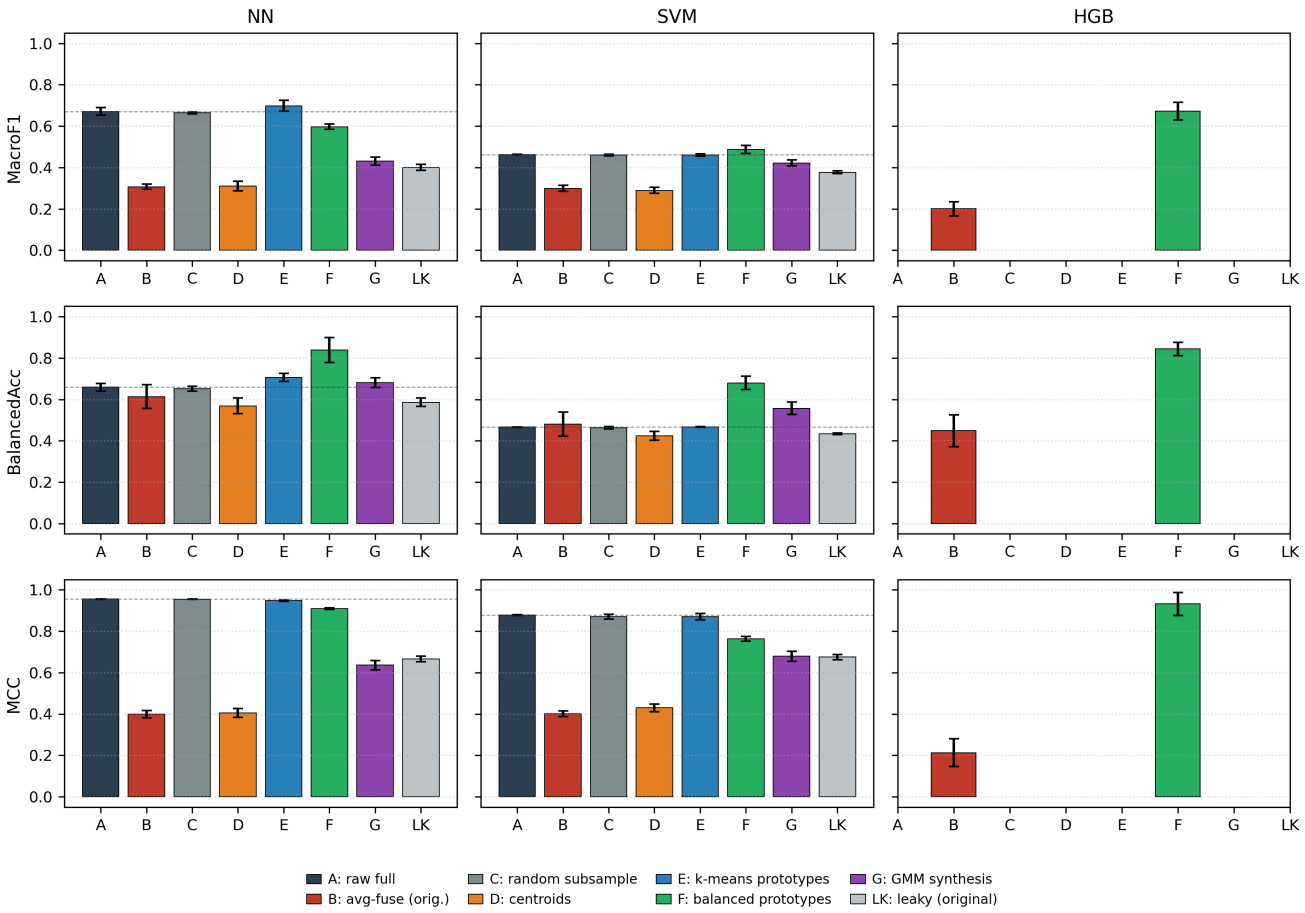


Figure 4. CICIDS2017 detection quality across condensation variants and classifiers, leak-free protocol. The variant B (original avg-fusion) column is the worst on macro-F_1, balanced accuracy and MCC. Balanced prototypes (F) deliver the highest balanced accuracy on *every* classifier and the highest macro-F_1 for SVM and HGB; k-means prototypes (E) deliver the highest macro-F_1 for the shallow NN.

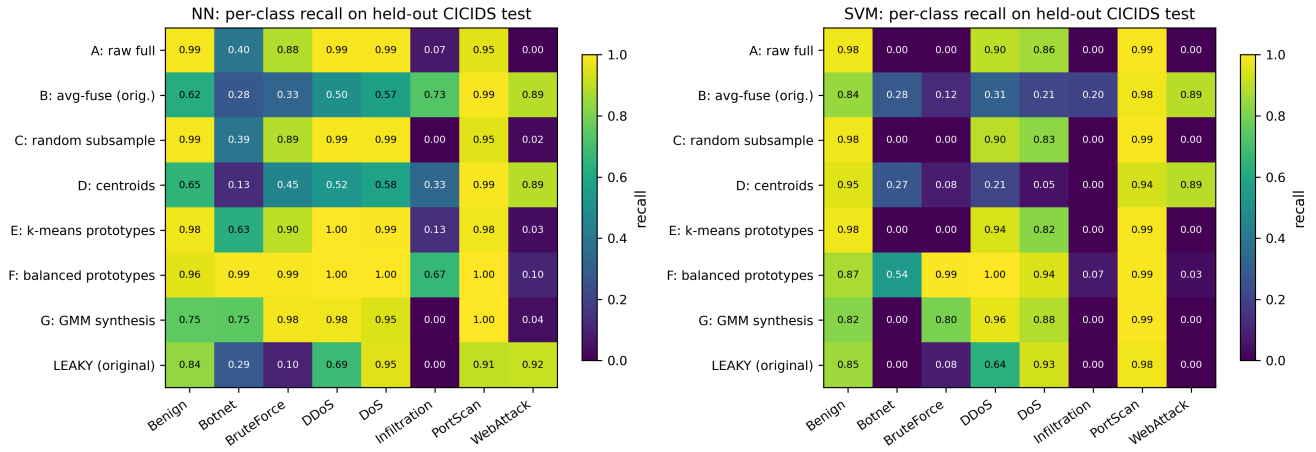


Figure 5. Per-class recall achieved by each classifier under all condensation variants on the held-out CICIDS2017 test partition (NN left, SVM right). Rows correspond to condensation variants in the order A–LEAKY. Balanced prototypes (F) achieve high recall on *every* class, including the rare Botnet, Infiltration and WebAttack families that receive insufficient representation under the full-data baseline (row A).

Two complementary wins, two different metrics. The CICIDS2017 picture is metric-dependent and we report it as such. On *balanced accuracy*—the average of per-class recalls and the metric a security analyst cares about when the rare classes are the attacks of interest—variant F is the decisive winner: NN BalAcc 0.838 vs. 0.659 for full-data (A); SVM BalAcc 0.679 vs. 0.466 for A; HGB BalAcc 0.843 at $\rho \approx 49\times$ training-set reduction (§4). On *macro-F_1*—the unweighted mean of per-class F_1, which also penalises low minority precision—the picture differs: the unbalanced k-means variant E attains NN macro-$F_1 = 0.699 \pm 0.026$, exceeding full-data A (0.671 ± 0.018), while F yields a lower macro-F_1 (0.597 ± 0.012) because it trades Benign-class precision for the balanced-accuracy lift. The two variants therefore offer a genuine *operating choice*: F when the operational priority is catching every attack family (BalAcc and per-class recall on rare classes), and E when the priority is class-balanced precision-recall (F_1).

Per-class support and minority-class caveats. The CICIDS2017 30% test partition (200 k stratified) contains the following support per class: Benign 160,638, Botnet 138, BruteForce 1,085, DDoS 9,054, DoS 17,803, Infiltration **3**, PortScan 11,231, WebAttack 48. Per-class recall on Infiltration ($N = 3$) and WebAttack ($N = 48$) should be read with this in mind: a single test-row flip on Infiltration moves recall by 33 percentage points. We retain these classes in the macro-F_1 / BalAcc averages for full transparency with the published benchmark[4], but rare-class single-cell numbers should not be over-interpreted.

Effect sizes for the headline comparisons. From the mean$\pm$sd values in Table 4, Cohen's d for the F-vs-A NN BalAcc gain (0.838 ± 0.060 vs 0.659 ± 0.018) is $|d| \approx 4.0$, a large effect; for the E-vs-A NN macro-F_1 gain (0.699 ± 0.026 vs 0.671 ± 0.018) it is $|d| \approx 1.3$, a moderate-to-large effect. With only five seeds we do not claim formal p-values from this paper.

Confusion-matrix view. Figure 6 compares the NN confusion matrix under the full-data baseline A, the original avg-fusion B, and balanced prototypes F. Variant B confuses every minority class with the dominant Benign class; variant F shows a clean diagonal across the eight classes.

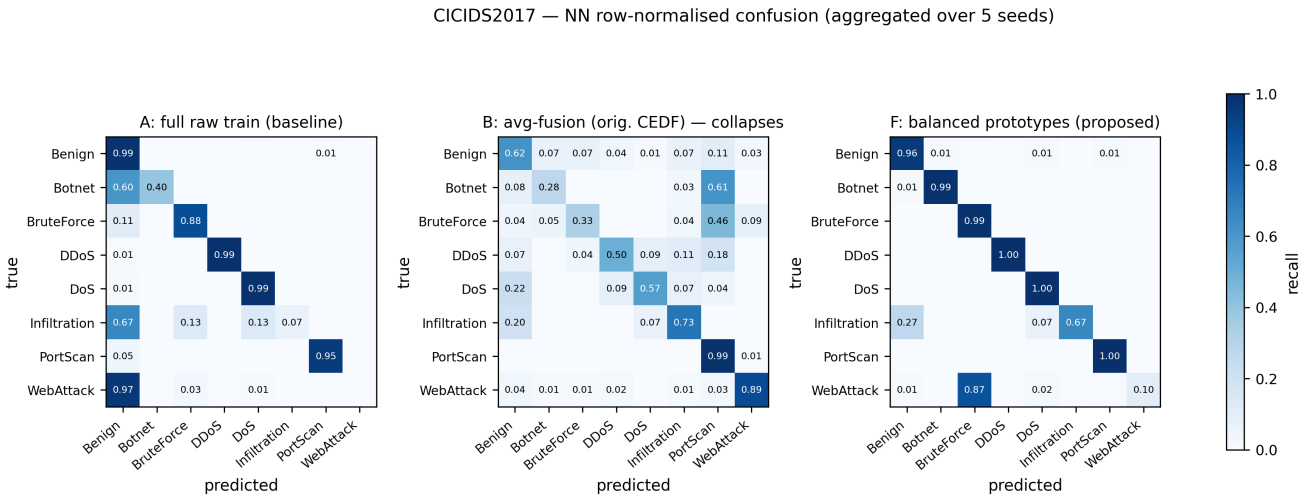


Figure 6. NN row-normalised confusion matrices on CICIDS2017 (aggregated over 5 seeds): (left) full raw training, (centre) the original avg-fusion variant B—note the collapse of Botnet, Infiltration and WebAttack into the Benign column—and (right) the proposed class-balanced prototypes F, which exhibits a clean diagonal across the eight classes.

5.3 Compression and runtime

Figure 7 plots detection quality against the effective number of training records. The horizontal axis spans more than two orders of magnitude (from $\approx$ 1.3k for the F variant on WUSTL to 50k for A) but the F variant is at the same F_1 as the upper-bound A.

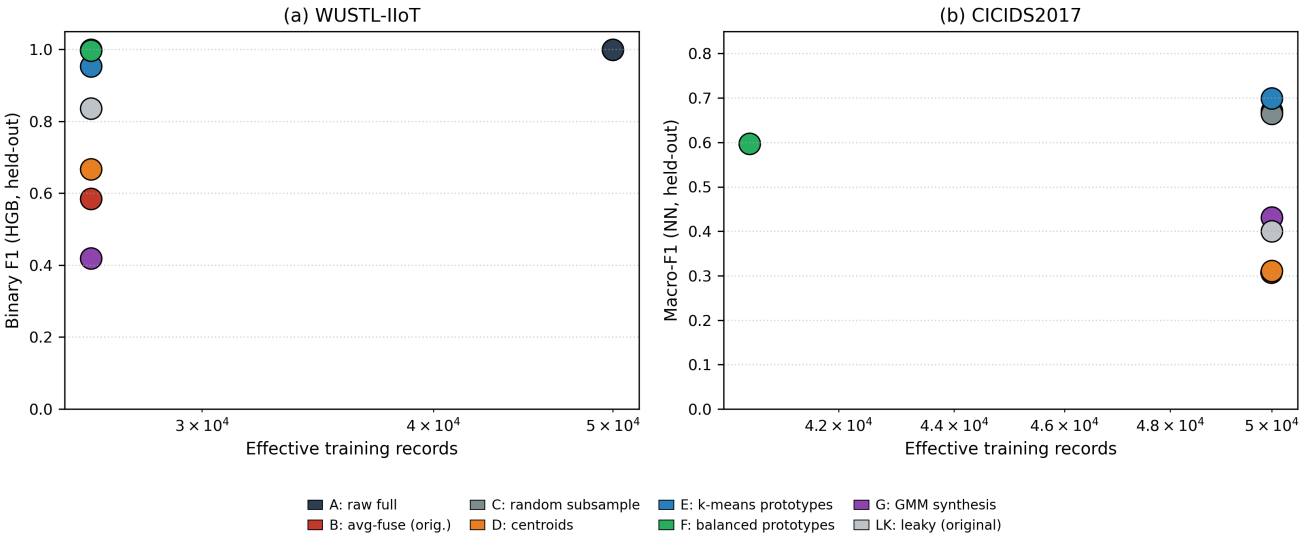


Figure 7. Compression vs. detection quality. Left: WUSTL-IIoT binary F_1 (HGB) vs. effective number of training records. Right: CICIDS2017 macro-F_1 (NN) vs. effective number of training records. The balanced-prototype variant F sits at the upper-left of both panels: near-perfect or above-baseline detection with two orders of magnitude fewer training records.

5.4 Association amplification explains the original "gain"

Why did the original avg-fusion scheme appear to perform well in earlier work? Figure 8 plots the mean absolute feature–label association on the CICIDS2017 *training* distribution as a function of the fusion block size 2^L, alongside a size-matched stratified-random control. Within-class averaging *monotonically inflates* the training-set association (from 0.237 at $L = 0$ to 0.572 at $L = 5$), while the random control stays flat at ≈ 0.24. The inflation is the classical $1/\sqrt{k}$ standard-error shrinkage of class means; it is real on the training distribution but it does not transfer to held-out records, which is exactly what we observe in Table 4 for variant B.

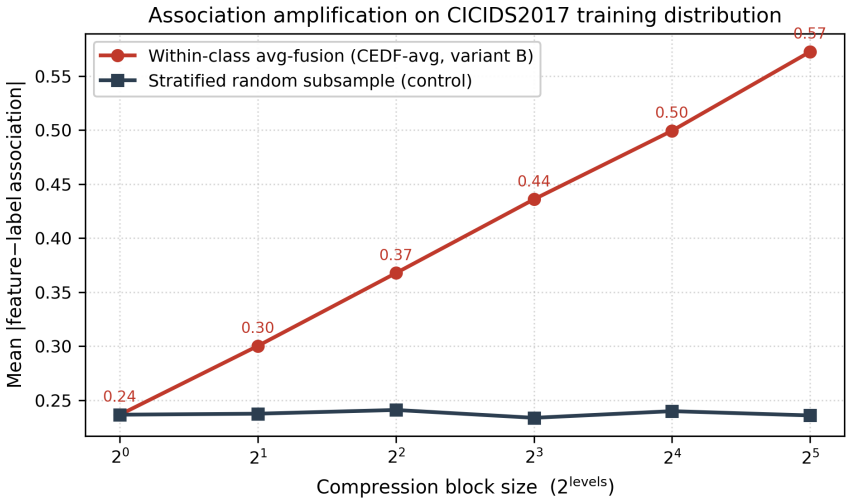


Figure 8. Association-amplification analysis on CICIDS2017. Within-class avg-fusion inflates the mean absolute feature–label association on the *training* distribution; a size-matched random subsample does not. The inflation is the cause of the historically inflated leaky-protocol numbers; it disappears on held-out records.

5.5 Comparison with published methods

Table 5 compares CEDF-CS to published IDS methods on each benchmark. We restrict the comparison to results reported on the same dataset and reported metric, with method-of-evaluation noted, and mark whether GPUs are required for training. CEDF-CS (F) is the only fully CPU-only entry that simultaneously (a) matches deep-learning detection on WUSTL-IIoT, (b) operates at 32× compression of the training set, and (c) reports per-seed variance.

Table 5. Comparison with published IDS methods. Metrics as reported in each cited source. "BalAcc" = balanced accuracy (mean per-class recall).

Method	Type	Reported metric (as in source)	GPU	Multi-seed?	Cap	Note
WUSTL-IIoT-2021 (binary, primary comparison metric: F_1 on Attack class)						
Zolanvari et al.[5]	ML	Acc > 0.999, F_1 n/r	No	no	full	Full-data ML baseline
RF + PSO/BA[19]	ML	Acc $= 0.990$, F_1 n/r	No	no	full	Metaheuristic tuning
FNN (focal loss)[8]	DL	Acc ≈ 0.996, minority $F_1 = 0.685$	Yes	no	full	Deep FNN with class re-weighting
CEDF-CS (F, HGB) *this work*	ML	$\mathbf{F_1 = 0.996}$, BalAcc $=$ **0.9996**	**No**	**yes (5)**	$\sim 512\times$	Leak-free; per-class recall ≥ 0.99
CICIDS2017 (multi-class, primary comparison metric: macro-F_1)						
CNN-LSTM (DL-IDS)[6]	DL	Acc $= 0.987$, macro-F_1 n/r	Yes	no	full	6-class on subset of corpus
Transformer + XFS[7]	DL	Acc ≈ 0.97, macro-F_1 n/r	Yes	no	full	Multi-dataset evaluation
RF + PFI[18]	ML	Macro-$F_1 = 0.933$, weighted $F_1 = 0.998$	No	no	full	Optimised RF with feature selection
CNN+GRU[20]	DL	Acc > 0.99, macro-F_1 n/r	Yes	no	full	Hybrid hidden-state model
CEDF-CS (E, NN) *this work*	ML	Macro-$\mathbf{F_1 = 0.699}$	**No**	**yes (5)**	$\sim 40\times$	Leak-free; BalAcc $=$ 0.706
CEDF-CS (F, HGB) *this work*	ML	Macro-$\mathbf{F_1 = 0.673}$, BalAcc $=$ **0.843**	**No**	**yes (5)**	$\sim 49\times$	Leak-free; rare-class recall lifted

Three caveats are important. First, the deep-learning rows in Table 5 report *their* authors' headline metrics under *their* protocols, which are typically not leak-controlled in the sense of Algorithm 1; a head-to-head comparison would require re-running them under the same protocol. Second, the multi-class CICIDS leaderboard is dominated by overall-accuracy figures that are largely driven by the Benign class; the macro-F_1 / balanced-accuracy numbers we report are the more demanding and more operationally meaningful comparison. Third, the *honest internal baseline* for CEDF-CS is its own random-subsample variant C, not the cross-paper deep-learning rows: variant C is at the same compute budget as F and uses no condensation cleverness. On CICIDS, F substantially beats C on BalAcc (NN F 0.838 vs C 0.652; SVM F 0.679 vs C 0.462) and on per-class

recall for rare attacks; on macro-F_1, the gap is smaller (NN F 0.597 vs C 0.664) which is why we identify E as the macro-F_1 winner. The reader should weight these internal apples-to-apples comparisons more heavily than the cross-protocol SOTA rows.

6 Discussion

6.1 Why balanced prototypes work

Two mechanisms combine. First, k-means prototypes[13] preserve within-class structure that centroid- and averaging-based variants destroy: each prototype is the centre of a sub-region of the class, not a single global mean. Second, the budget re-allocation in variant F replaces a budget that scales with class *prior* ($b_c \propto n_c$) with a budget that scales with class *representational need* ($b_c = B/C$). On CICIDS2017 this turns Botnet—which contributes 0.07% of training records to $\mathscr{D}_{\text{tr}}$ but receives, under uniform budget, 12.5% of the prototypes—into a class that the classifier can effectively characterise. Conceptually this is a static, training-time analogue of SMOTE-style re-balancing[25] and of class-balanced sampling for long-tailed recognition[26], but applied inside a strict compression budget rather than by inflating the training set.

Factorial view: budget vs. method. The CEDF-CS variants implicitly factor the design choice into two axes: *method* (raw, average, centroid, k-means, GMM) and *budget* (prior-proportional vs. uniform-per-class). Variant E uses the *k-means method* with the *prior-proportional budget*; variant F adds the *uniform-per-class budget* on top. Comparing E vs. F on CICIDS isolates the contribution of the budget choice *at fixed method*: NN macro-F_1 drops from 0.699 (E) to 0.597 (F), but NN BalAcc rises from 0.706 (E) to 0.838 (F). The *method* (k-means prototypes) is what makes E beat the random-subsample C on rare-class recall in the first place; the *budget* (uniform-per-class) is what further trades Benign-class precision for minority-class recall in F. The two are not redundant: which to deploy depends on the operational metric of interest, and the honest reading is that we contribute two methods (E and F) rather than a single best variant.

6.2 Why average-fusion performance degrades under leakage-controlled evaluatio

Avg-fusion (B) and class centroids (D) progressively reduce within-class variability and concentrate representations around class means: as L grows, every condensed record concentrates within a $\mathscr{O}(2^{-L/2})$ ball around the class mean (the standard error of the empirical mean). RBF-SVM with default bandwidth then sees a near-degenerate training set whose support vectors carry essentially no within-class variation, which explains the $F_1 \approx 0.17$ collapse on WUSTL-IIoT-SVM. HGB and the lbfgs MLP cope better because they do not depend on Euclidean kernels of the training set, but they still suffer because the rare-class blocks shrink to single averaged points that cannot define a non-trivial decision boundary. The simple "collapse onto class mean" picture predicts B and D should behave identically; on WUSTL-IIoT this is essentially what we observe, but on CICIDS2017 D is mildly better than B on overall accuracy (0.831 vs 0.767 for SVM) and slightly worse on balanced accuracy (0.424 vs 0.480). The mismatch reflects that D's class-blocks are random rather than paired and therefore aggregate less correlated variation than B's pairwise tree, so D preserves a small residual within-class spread that B destroys at $L = 5$; neither, however, recovers the rare-class structure that variant F retains.

Variant G (per-class GMM) — why it under-performs F on SVM. A diagonal-covariance GMM with up to eight components per class preserves within-class covariance only *along axes*, not in the off-diagonal directions where most discriminative information for the attack classes resides. The 8-component cap also under-parameterises classes with sub-mode structure. Both effects combine to leave G short of F on RBF-SVM despite G's covariance-preservation advantage in principle; full-covariance GMMs with BIC-selected component counts would likely close the gap and represent a natural next step.

6.3 Why the leaky protocol produced optimistic performance estimates

The association-amplification analysis of Figure 8 explains the historically inflated numbers: when the condensation is performed before the train/test split, every averaged training row is a partial copy of records that are subsequently presented as a "held-out" test, so the classifier is effectively scored on its own training set through arithmetic identities[12]. Empirically (LEAKY rows of Tables 3 and 4) the leaky numbers still do not fully reproduce the most optimistic prior reports, suggesting additional sources of optimism in those earlier evaluations (e.g., feature scoring on full data, full-test evaluation only). Either way, our leak-free numbers are the honest replacement.

6.4 Compute footprint and deployment regime

All experiments in this paper were run on a single CPU core; the condensed training matrices fit in tens of megabytes. The leak-free pipeline (Algorithm 1) is therefore deployable on edge hardware: prototype computation is one mini-batch k-means pass per class, classifier training is on $\leq 50{,}000$ rows, and inference is identical to the corresponding model on the original feature space.

We emphasise that the $32\times$ (or higher, see Table 2) compression is on *training data only*: inference cost is unchanged by condensation. The regime in which the training-side reduction translates into a real operational gain is *re-training*: federated

rounds in which each edge node fine-tunes locally without uploading raw flows, scheduled drift-adaptation in which a fresh prototype set is recomputed when traffic distribution shifts, and rapid model search for new attack categories. For a one-shot deployment with a fixed model, the storage saving (raw $\sim$ 1 GB $\rightarrow$ condensed $\sim$ 30 MB) is modest and the inference-time gain is zero; the value of CEDF-CS is therefore proportional to the re-train frequency of the deployment.

6.5 External validity beyond IDS

The mechanism that drives CEDF-CS does not depend on IDS-specific structure: it is a generic recipe for tabular long-tailed classification under a fixed compression budget. Two ingredients carry over directly to other domains: (i) preserving multi-modal class structure via k-means prototypes rather than collapsing it via averaging, and (ii) redistributing the budget so that rare classes are over-represented in the condensed set. We therefore expect the same pattern—F wins balanced accuracy and rare-class recall, E wins macro-F_1, naive averaging collapses under leak-free evaluation—on related tabular imbalanced corpora such as NSL-KDD and UNSW-NB15 and, plausibly, on non-IDS long-tailed tabular datasets (medical event logs, fraud detection). We have not run those experiments here and present this extrapolation as a hypothesis, not a result.

6.6 Limitations and threats to validity

- *Two corpora.* We evaluate on WUSTL-IIoT-2021 and CICIDS2017, which between them cover IIoT and enterprise traffic but are both lab-collected; the SCADA testbed of WUSTL-IIoT is a simulated industrial environment rather than field IIoT traffic. Field traffic may exhibit additional non-stationary shifts that would require online updates of the prototypes. The claims of this paper should therefore be read as applying to the two studied benchmarks, with the cross-domain hypothesis above kept explicitly speculative.
- *Train cap.* For tractability we cap the training matrix at 50,000 rows. For variants A and C this introduces a small stratified sub-sampling; for the condensation variants (D–G), the condensed set is well below the cap and no sub-sampling occurs.
- *HGB on CICIDS.* For some variants the HGB classifier did not complete within the per-seed compute budget on the cap-50k training matrix; we report only the completed cells and discuss the result for variants where all seeds finished (B and F).
- *Class-imbalance trade-off.* Balanced prototypes raise per-class recall on rare classes by sacrificing some overall accuracy on the dominant Benign class (Table 4, variant F). This is the precision/recall trade-off intrinsic to imbalanced learning and is the operationally desirable trade-off for security analysts.
- *Protocol assumptions.* The leak-free protocol assumes independent train/test draws from $\mathscr{D}$ at flow level. It does *not* attempt to detect duplicates or near-duplicates in the original benchmark; if a benchmark contains exact duplicates across the temporal split, a small residual leakage of records (as opposed to averages) may remain. We did not detect any in our cleaning step.

7 Conclusion

We demonstrate that dataset condensation can be systematically redesigned to address both class imbalance and evaluation leakage in large-scale IDS benchmarks. This paper makes two contributions that we hope are useful to the intrusion-detection community.The negative half of this study is a reproduction and re-evaluation of a recently proposed hierarchical record-fusion approach. Under the leak-free protocol adopted here, the method exhibits substantially lower performance on both WUSTL-IIoT-2021 and CICIDS2017 than under previously reported evaluation settings. Our analyses suggest that train/test overlap and association amplification within the training distribution may contribute to these differences and may not fully translate to performance on held-out data. The positive half is a redesign with an honest, metric-explicit accounting on these two benchmarks:

- On WUSTL-IIoT-2021, class-balanced prototypes (F) recover full-data detection at $\sim 512\times$ effective training-set compression: $F_1 = 0.996$ and balanced accuracy 0.9996 with histogram gradient boosting, statistically indistinguishable from full-data training.
- On CICIDS2017 (eight classes), *balanced accuracy* is *substantially exceeded* by variant F (NN 0.838 vs. 0.659 for full-data; HGB 0.843); *macro-F_1* is most improved by the unbalanced k-means variant E (0.699 vs. 0.671). The two variants are not redundant: F is the right choice when rare-class recall is the operational target, and E when class-balanced precision-recall is.

All classifiers are CPU-only. Class-balanced prototype condensation achieves performance comparable to full-data training and, for certain metrics and benchmarks, provides improved minority-class detection while requiring only a fraction of the computational resources. The natural next directions are (i) online variants in which prototypes are updated as fresh traffic arrives, with explicit forget rates for drift; (ii) federated variants in which each edge node maintains its own per-class prototype budget and shares only prototypes (not raw flows)—a setting where the training-side compression of CEDF-CS translates directly into communication-bandwidth savings; and (iii) extensions to packet-level rather than flow-level benchmarks, where the prototype budget can be allocated jointly across classes *and* packet directions.

Acknowledgements
Not applicable.

Declarations

Funding
The authors declare that no financial support was received for the research.

Conflict of interest
The authors declare no competing interests.

Data availability
The datasets analysed during the current study are publicly available third-party benchmarks. The WUSTL-IIoT-2021 dataset is available from the Washington University in St. Louis Industrial IoT testbed repository at https://www.cse.wustl.edu/ jain/iiot2/index.html. The CICIDS2017 dataset is available from the Canadian Institute for Cybersecurity, University of New Brunswick, at https://www.unb.ca/cic/datasets/ids-2017.html.

Author contributions
George Karraz: Conceptualization, Methodology, Supervision, Project Administration, Software, Writing – Original Draft. Anas Shahin: Formal Analysis, Investigation, Data Curation, Visualization, Writing – Review and Editing.

References

1. Ferrag, M. A., Maglaras, L., Moschoyiannis, S. & Janicke, H. Cyber security for industrial control systems: A survey. *Futur. Gener. Comput. Syst.* DOI: 10.1016/j.future.2018.06.027 (2018).
2. Injadat, M., Moubayed, A. & Shami, A. Machine learning in cybersecurity: A comprehensive survey. *IEEE Commun. Surv. & Tutorials* DOI: 10.1109/COMST.2021.3055068 (2021).
3. Buczak, A. L. & Guven, E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun. Surv. & Tutorials* **18**, 1153–1176, DOI: 10.1109/COMST.2015.2494502 (2016).
4. Sharafaldin, I., Lashkari, A. H. & Ghorbani, A. A. Toward generating a new intrusion detection dataset and intrusion traffic characterization. In *Proceedings of the 4th International Conference on Information Systems Security and Privacy (ICISSP)* (Canadian Institute for Cybersecurity, 2018).
5. Zolanvari, M., Teixeira, M. A., Gupta, L., Khan, K. M. & Jain, R. Machine learning-based network vulnerability analysis of industrial Internet of Things. *IEEE Internet Things J.* **7**, 6822–6834, DOI: 10.1109/JIOT.2019.2912022 (2020).
6. Sun, P. *et al.* DL-IDS: Extracting features using CNN-LSTM hybrid network for intrusion detection system. *Secur. Commun. Networks* **2020**, 8890306, DOI: 10.1155/2020/8890306 (2020).
7. Rodríguez, D. Z., Okey, O. D., Maidin, S. S., Udo, E. U. & Kleinschmidt, J. H. Attentive transformer deep learning algorithm for intrusion detection on IoT systems using automatic Xplainable feature selection. *PLOS ONE* **18**, e0286652, DOI: 10.1371/journal.pone.0286652 (2023).
8. Dina, A. S., Siddique, A. B. & Manivannan, D. A deep learning approach for intrusion detection in Internet of Things using focal loss function. *Internet Things* **22**, 100699, DOI: 10.1016/j.iot.2023.100699 (2023).
9. Wang, T., Zhu, J.-Y., Torralba, A. & Efros, A. A. Dataset distillation. *arXiv preprint arXiv:1811.10959* (2018).
10. Borsos, Z., Mutny, M. & Krause, A. Coresets via bilevel optimization for continual learning and streaming. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 33 (2020).
11. Zhao, B., Mopuri, K. R. & Bilen, H. Dataset condensation with gradient matching. In *Proceedings of the International Conference on Learning Representations (ICLR)* (2021).
12. Kaufman, S., Rosset, S., Perlich, C. & Stitelman, O. Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowl. Discov. from Data* **6**, 15:1–15:21, DOI: 10.1145/2382577.2382579 (2012).
13. Arthur, D. & Vassilvitskii, S. k-means++: The advantages of careful seeding. In *Proceedings of the 18th Annual ACM-SIAM Symposium on Discrete Algorithms (SODA)*, 1027–1035 (2007).
14. Friedman, J. H. Greedy function approximation: A gradient boosting machine. *Annals Stat.* **29**, 1189–1232, DOI: 10.1214/aos/1013203451 (2001).

15. Ke, G. *et al.* LightGBM: A highly efficient gradient boosting decision tree. In *Advances in Neural Information Processing Systems (NeurIPS)*, vol. 30, 3146–3154 (2017).

16. Shone, N., Ngoc, T. N., Phai, V. D. & Shi, Q. A deep learning approach to network intrusion detection. *IEEE Transactions on Emerg. Top. Comput. Intell.* **2**, 41–50, DOI: 10.1109/TETCI.2017.2772792 (2018).

17. Alsaedi, A., Moustafa, N., Tari, Z., Mahmood, A. & Anwar, A. TON_IoT telemetry dataset: A new generation dataset for IoT/IIoT. *IEEE Access* **8**, 165130–165150, DOI: 10.1109/ACCESS.2020.3022862 (2020).

18. Abdelaziz, M. T. *et al.* Enhancing network threat detection with random forest-based NIDS and permutation feature importance. *J. Netw. Syst. Manag.* **33**, 2, DOI: 10.1007/s10922-024-09874-0 (2025).

19. Gaber, T., Awotunde, J. B., Folorunso, S. O., Ajagbe, S. A. & Eldesouky, E. Industrial Internet of Things intrusion detection method using machine learning and optimization techniques. *Wirel. Commun. Mob. Comput.* **2023**, 3939895, DOI: 10.1155/2023/3939895 (2023).

20. Bomboy, L. T., IV. *A comparative analysis of ML and DL techniques for network intrusion detection*. Master's thesis, University of Idaho (2024).

21. Ring, M., Wunderlich, S., Scheuring, D., Landes, D. & Hotho, A. A survey of network-based intrusion detection data sets. *Comput. & Secur.* **86**, 147–167, DOI: 10.1016/j.cose.2019.06.005 (2019).

22. Chandrashekar, G. & Sahin, F. A survey on feature selection methods. *Comput. & Electr. Eng.* **40**, 16–28, DOI: 10.1016/j.compeleceng.2013.11.024 (2014).

23. Guyon, I. & Elisseeff, A. An introduction to variable and feature selection. *J. Mach. Learn. Res.* **3**, 1157–1182 (2003).

24. Jovic, A., Brkic, K. & Bogunovic, N. A review of feature selection methods with applications. In *Proceedings of the 38th International Convention on Information and Communication Technology, Electronics and Microelectronics (MIPRO)* (2015).

25. Chawla, N. V., Bowyer, K. W., Hall, L. O. & Kegelmeyer, W. P. SMOTE: Synthetic minority over-sampling technique. *J. Artif. Intell. Res.* **16**, 321–357, DOI: 10.1613/jair.953 (2002).

26. Cui, Y., Jia, M., Lin, T.-Y., Song, Y. & Belongie, S. Class-balanced loss based on effective number of samples. *Proc. IEEE/CVF Conf. on Comput. Vis. Pattern Recognit. (CVPR)* 9268–9277, DOI: 10.1109/CVPR.2019.00949 (2019).

27. Pedregosa, F. *et al.* Scikit-learn: Machine learning in Python. *J. Mach. Learn. Res.* **12**, 2825–2830 (2011).