Beyond Accuracy: A Leakage-Controlled and Operationally Grounded Evaluation of Machine Learning for IoT Botnet Detection in Resource-Constrained Organisations

Thivyah A/P Vijayan, Farhad Nadi, Mohd Fahmi Bin Mohd Nordin*

Faculty of Artificial Intelligence & Frontier Technologies, UNITAR International University, Petaling Jaya, Selangor, Malaysia

***Corresponding author**

DOI: https://doi.org/10.47772/IJRISS.2026.101000400

Received: 21 October 2026; Accepted: 26 October 2026; Published: 03 November 2026

ABSTRACT

Internet of Things (IoT) devices are widely deployed in households, small and medium-sized enterprises (SMEs) and public bodies, many of which lack the specialist capacity to investigate intrusion alarms. Studies on benchmark datasets frequently report near-perfect detection accuracy, yet such figures may not reflect performance in practice. This study aimed to evaluate machine learning detectors for IoT botnet traffic under leakage controls and in terms of the alarm workload they would impose. Random Forest, Decision Tree and radial-basis Support Vector Machine classifiers were trained on the official UNSW-NB15 and Bot-IoT partitions after identifier and label-derived attributes had been removed, with features selected by Mutual Information, alarm thresholds chosen on a validation hold-out and duplicate-aware cross-validation. Performance was assessed with bootstrap confidence intervals, McNemar tests, permutation importance and a base-rate analysis of alarm precision. In UNSW-NB15, 38.6% of test flows duplicated a training flow on the selected features. When the excluded attributes were restored, the Decision Tree classified every test flow correctly on both datasets, confirming target leakage. Without leakage, Random Forest performed best on UNSW-NB15 (F1-score 0.902; area under the receiver operating characteristic curve 0.973) but misclassified 20.2% of benign flows. A validation-chosen threshold reduced this false-positive rate to 2.2% while still detecting 80.6% of attacks, falling to 57.7% for test flows absent from training on these features; at an attack prevalence of 1%, only about 27% of alarms would be genuine. Bot-IoT was nearly perfectly separable, but its 107 benign test flows allowed only an imprecise false-positive estimate (Random Forest: 4.7%; 95% confidence interval 2.0%–10.5%). These findings indicate that reported accuracy is a poor guide to the operational value of IoT intrusion detection. Evaluations and product claims should report false-positive rates and alarm precision at realistic prevalence, and such detectors are better suited to triage than to autonomous response.

Keywords: IoT botnet detection, machine learning, false alarms, data leakage, resource-constrained organisations

INTRODUCTION

The Internet of Things (IoT) connects large numbers of inexpensive devices that often run with limited processing power, weak or default credentials and rarely updated firmware (Alaba et al., 2017; Hussain et al., 2020). Botnets such as Mirai showed how quickly such devices can be compromised at scale and turned into infrastructure for distributed denial-of-service (DDoS) attacks (Antonakakis et al., 2017). Households, small and medium-sized enterprises (SMEs) and local public bodies operate a growing share of these devices but seldom employ security analysts. For them, the value of an intrusion detection system (IDS) depends less on its reported accuracy than on whether its alarms can be understood and acted upon: false alarms consume analyst time and erode trust (Alahmadi et al., 2022), and machine learning has long struggled to move from laboratory evaluations into operational monitoring (Sommer & Paxson, 2010).

Studies that apply machine learning to the public UNSW-NB15 and Bot-IoT benchmarks frequently report accuracies close to 100% (Aljohani et al., 2024; El-Sofany et al., 2024; Pokhrel et al., 2021). Such figures warrant caution. Security evaluations are prone to data snooping, spurious correlations and inappropriate measures (Arp et al., 2022); label information can leak into the input features (Kaufman et al., 2012); and, because attacks are rare in real networks, even a small false-positive rate can mean that most alarms are false (Axelsson, 2000). This paper re-examines the evidence with a leakage-controlled and operationally grounded evaluation of Random Forest (RF), Decision Tree (DT) and Support Vector Machine (SVM) classifiers on the official partitions of both datasets. It addresses four research questions:

- **RQ1.** How do target leakage and record duplication in the benchmark partitions affect reported detection performance?
- **RQ2.** How do RF, DT and SVM compare when evaluated without leakage and with the uncertainty of the estimates quantified?
- **RQ3.** What alarm workload and alarm precision would these detectors impose on a resource-constrained organisation, and how far can the choice of decision threshold reduce that burden?
- **RQ4.** Which traffic features drive the best model's decisions, and how can they be communicated to non-specialist decision makers?

The study makes four contributions: (i) an experimental demonstration that identifier and label-derived attributes can produce perfect scores on these datasets, with a quantification of duplication, conflicting labels and train–test overlap; (ii) a leak-free comparison of the three models with confidence intervals, McNemar tests, nested and duplicate-aware cross-validation, and evaluation on test flows absent from training; (iii) a translation of detection rates into alarm precision and daily false-alarm volumes across validation-chosen alarm budgets; and (iv) model-agnostic explanations and inference costs relevant to low-resource deployment. All code, per-flow predictions and a script that recomputes every reported figure are provided as supplementary material.

LITERATURE REVIEW

IoT botnets, benchmarks and detectors

Surveys identify resource-constrained nodes, weak authentication and poor update practices as root causes of IoT compromise (Alaba et al., 2017; Hussain et al., 2020), and systematic reviews of botnet detection note persistent problems with interpretability, false positives and adaptation to new threats (Gelgi et al., 2024; Liu et al., 2025). UNSW-NB15 provides normal traffic and nine attack families described by 49 flow features (Moustafa & Slay, 2015), and Bot-IoT contains simulated IoT traffic with DoS, DDoS, reconnaissance and information-theft attacks, released with a ten-best-feature version and predefined training and testing files (Koroniotis et al., 2019). On these benchmarks, Random Forest (Breiman, 2001) is repeatedly found to be accurate (Aljohani et al., 2024; El-Sofany et al., 2024), and oversampling with SMOTE (Chawla et al., 2002) is widely used (Pokhrel et al., 2021). Efficiency is recognised as a concern for edge deployment (Wakili et al., 2025) but is rarely reported alongside detection metrics.

Evaluation pitfalls

Class imbalance makes accuracy misleading, and precision-recall analysis is more informative than ROC analysis for the minority class (Saito & Rehmsmeier, 2015). Leakage occurs when features encode the target or the data-collection process (Kaufman et al., 2012); Arp et al. (2022) found such pitfalls widespread in security research, and Kapoor and Narayanan (2023) documented leakage across 17 fields, including preprocessing fitted before the train–test split and duplicates spanning both partitions. When attacks are rare, even a low false-positive rate yields mostly false alarms (Axelsson, 2000). The benchmarks themselves can be flawed: Engelen et al. (2021) relabelled more than 20% of CICIDS2017 traces, and Flood et al. (2024) identified recurring design flaws in seven intrusion detection datasets. Because each benchmark uses its own features, Sarhan et al. (2022) proposed common NetFlow feature sets; using them, Layeghy and Portmann (2023) found that no model generalised across all datasets studied.

Positioning of the present study

Table 1 contrasts the present study with representative work. Applied studies report near-perfect results without describing the removal of identifier or label-derived attributes, duplicate handling or uncertainty estimates; methodological studies identify these problems but do not quantify their combined effect on IoT botnet detection or translate it into the alarm workload of small adopters. This study addresses that gap.

Table 1: Comparison of the present study with related work

Study	Focus and main reported outcome	Relation to the present study
El-Sofany et al. (2024); Pokhrel et al. (2021)	ML classifiers with SMOTE on UNSW-NB15 or Bot-IoT; accuracy above 99%	No reported removal of label-derived attributes, duplicate audit or confidence intervals
Arp et al. (2022); Kapoor & Narayanan (2023)	Leakage, data snooping and base-rate pitfalls in published studies	Pitfalls tested empirically on IoT botnet benchmarks
Engelen et al. (2021); Flood et al. (2024)	Labelling and design flaws in benchmark datasets	Complemented by a duplicate, label-conflict and leakage audit
Sarhan et al. (2022); Layeghy & Portmann (2023)	Common NetFlow features; poor cross-dataset generalisation	Cross-dataset evaluation identified as the next step
Apruzzese et al. (2023)	Pragmatic assessment of NIDS configurations and hardware	Adds alarm precision at realistic prevalence and alarm budgets
Present study	Leakage ablation, duplicate-aware nested CV, unseen-flow evaluation, threshold sweep, base-rate analysis	—

METHODOLOGY

Research design and datasets

The study adopts a quantitative, comparative experimental design using secondary benchmark data, with criteria chosen from the perspective of a resource-constrained adopter: detection, false-alarm volume, explainability and computational cost. The pipeline, applied identically to both datasets except where noted, proceeds in six steps: removal of identifier and label-derived attributes; a duplicate and leakage audit; a validation hold-out; feature selection and class balancing on the fitting data; training of the three classifiers; and evaluation on the official test partitions with statistical and operational analysis. The official training and testing partitions released by the dataset authors were used without modification: for UNSW-NB15 (Moustafa & Slay, 2015), 175,341 training and 82,332 test records (37,000 normal in the test partition); for the ten-best-feature version of Bot-IoT (Koroniotis et al., 2019), 2,934,817 training and 733,705 test records, of which only 370 and 107 were normal. Flows were classified as normal (0) or attack (1).

Data preparation and leakage control

Attributes that identify individual records or restate the class label were removed before analysis. In UNSW-NB15 these were the record identifier and the attack category; in Bot-IoT they were the record sequence numbers, the addresses of the testbed's attacking and victim hosts, and the attack category and subcategory. Categorical features were encoded using only the categories observed in training, and port numbers were expressed numerically; neither dataset contained missing values. Duplicates were defined as records with identical feature vectors, both over the features retained by Mutual Information, which the models receive, and over all candidate features. The audit counted repeated records, training records whose vector also occurs with the opposite label (conflicting labels) and test records whose vector occurs in training. For the main analysis, duplicates were retained so that results refer to the partitions as distributed; their influence was assessed by evaluating on test flows absent from training, by re-training on deduplicated data, and by cross-validation with identical vectors kept in one fold. A stratified 15% validation split was held out from each training partition and used only to choose thresholds. Table 2 lists the data on which every estimated step was fitted; the test partitions were used only to score the final models.

Table 2: Leakage-control protocol: data used to fit each step

Step	Fitted on	Applied to
Category encoding	Training partition	Validation and test data (categories absent from training assigned a reserved code)
Mutual Information feature ranking	Fitting split (main models); training folds (cross-validation)	All splits, using the same selected features
SMOTE (UNSW-NB15 only)	Fitting split; training folds	Never applied to validation, test or held-out folds
Standardisation (SVM)	SVM training sample	Validation and test data
Model parameters	Fitting split (main models); training folds (cross-validation)	—
Low-alarm and budget thresholds	Validation split	Test partition

Features, classifiers and operating thresholds

Features were ranked by Mutual Information with the label on up to 100,000 fitting records (retaining all normal Bot-IoT records); the top 20 were kept for UNSW-NB15 and the top 10 of 12 candidates for Bot-IoT. SMOTE was applied to the UNSW-NB15 fitting split only (101,439 records per class); Bot-IoT, with 314 normal fitting records, used class weighting for the SVM instead. The classifiers were used with standard default settings, reflecting what a small organisation without tuning expertise would deploy: a Random Forest of 100 trees, an unpruned decision tree using the Gini criterion, and an SVM with a radial basis function kernel and balanced class weights applied to standardised features. The SVM was trained on 20,000 fitting records because the cost of kernel training grows faster than linearly with sample size. Each model was evaluated at the default threshold (probability 0.5; SVM decision value 0) and at a low-alarm threshold, the smallest whose validation false-positive rate did not exceed 1%, mirroring an alarm budget set in advance. Budgets of 0.1%, 0.5%, 2% and 5% were also examined for RF and SVM on UNSW-NB15; the DT was excluded because its scores are almost binary, and Bot-IoT because its validation split has only 56 normal records.

Evaluation and statistical analysis

With attack as the positive class, precision, recall (true-positive rate, TPR), F1, false-positive rate (FPR), balanced accuracy, macro-F1 and ROC-AUC were computed on the test partitions, with stratified bootstrap 95% intervals (2,000 resamples) and Wilson intervals for FPR (Wilson, 1927). Models were compared with McNemar's test (McNemar, 1947; Dietterich, 1998). Five-fold cross-validation on the training partitions refitted Mutual Information selection and SMOTE on each training fold, with stratified group folds keeping records with identical vectors over all candidate features together. Because the selected features are unknown before a fold is fitted, a sensitivity analysis grouped on the features selected on the fitting split, and two stratified designs were run for comparison. For Bot-IoT, false positives were pooled over the held-out folds, which contain all 370 normal training records, to give a second FPR estimate. Test metrics were also computed on flows whose vectors do not occur in training. Alarm precision (positive predictive value, PPV) was computed for attack prevalences p of 10%, 1% and 0.1% as $\mathrm{PPV} = \mathrm{TPR}\cdot p / (\mathrm{TPR}\cdot p + \mathrm{FPR}\cdot(1 - p))$ (Axelsson, 2000), and daily false alarms for an illustrative 100,000 benign flows. Explanations used RF permutation importance, and a leakage ablation re-trained RF and DT with the excluded attributes restored.

Experiments used Python 3.14.4, scikit-learn 1.9.1 (Pedregosa et al., 2011) and imbalanced-learn 0.14.2 (Lemaître et al., 2017) with a fixed seed. A verification script recomputes every metric, interval, test, subset result and threshold-sweep value from the released per-flow outputs; all 611 checks agreed, and re-running the pipeline from the raw data reproduced every metric exactly. The code, per-flow outputs and verification script are provided as supplementary material. Only public laboratory datasets without personal data were used.

RESULTS

Data audit and leakage (RQ1)

Table 3 summarises the audit. Once identifiers were removed, duplication was substantial in UNSW-NB15. On

the 20 selected features, 49.6% of training records repeated an earlier vector and 31,784 test records (38.6%) also occurred in training, most of them attacks. A further 7,892 training records, sharing 283 vectors, occurred with both labels, which limits the accuracy attainable from these features. On all 42 candidate features the overlap was smaller (10.4%), showing that feature selection merges flows that differ only in discarded attributes. Bot-IoT showed little overlap (0.47%) and no conflicting labels.

Table 3: Data audit of the official partitions

Audit item	**UNSW-NB15 (selected features)**	**UNSW-NB15 (all candidate features)**	**Bot-IoT**
Exact duplicates with identifiers present	0 / 0		0 / 0
Duplicate training records	87,015 (49.6%)	74,301 (42.4%)	7,793 (0.27%)
Duplicate test records	36,287 (44.1%)	28,386 (34.5%)	511 (0.07%)
Training records with conflicting labels	7,892	940	0
Test records also present in training	31,784 (38.6%)	8,541 (10.4%)	3,484 (0.47%)
of which normal / attack	5,070 / 26,714	1,472 / 7,069	6 / 3,478
matching a training record of the opposite class	4,295	524	0

Note: No missing values were found. Bot-IoT results are identical under both definitions; its 10 selected features are 10 of 12 candidates.

With the excluded attributes restored, the Decision Tree classified every test record correctly on both datasets and RF was perfect on Bot-IoT (Table S1). Because the attack-category attributes encode the class label, a single split separates the classes, so perfect scores of this kind indicate leakage rather than detection ability. Without these attributes, balanced accuracy on UNSW-NB15 fell to 0.858 for DT and 0.877 for RF.

Detection performance (RQ2)

Table 4: Test-set performance at the default threshold (95% confidence intervals in brackets)

Dataset	**Model**	**Precision**	**Recall**	**F1**	**FPR**	**Balanced accuracy**	**Macro-F1**	**ROC-AUC**
UNSW-NB15	RF	0.853	0.956	0.902	0.202	0.877	0.882	0.973
				[0.900, 0.903]	[0.198, 0.206]	[0.875, 0.879]		[0.972, 0.974]
UNSW-NB15	DT	0.844	0.926	0.883	0.21	0.858	0.862	0.869
				[0.881, 0.885]	[0.205, 0.214]	[0.856, 0.861]		[0.867, 0.871]
UNSW-NB15	SVM	0.76	0.978	0.855	0.379	0.8	0.804	0.926
				[0.853, 0.857]	[0.374, 0.384]	[0.797, 0.802]		[0.924, 0.928]
Bot-IoT	RF	1	1	1	0.047	0.977	0.988	1
				[1.000, 1.000]	[0.020, 0.105]	[0.958, 0.995]		[1.000, 1.000]
Bot-IoT	DT	1	1	1	0.056	0.972	0.974	0.972

				[1.000, 1.000]	[0.026, 0.117]	[0.949, 0.991]		[0.949, 0.991]
Bot-IoT	SVM	1	0.996	0.998	0	0.998	0.533	0.999
				[0.998, 0.998]	[0.000, 0.035]	[0.998, 0.998]		[0.999, 0.999]

Note: Attack is the positive class. FPR intervals are Wilson score intervals; others are stratified bootstrap percentile intervals. Confusion matrices and ROC and precision-recall curves are given in Figure S2 and Figure S3.

On UNSW-NB15 (Table 4), RF achieved the highest F1 (0.902), balanced accuracy (0.877) and ROC-AUC (0.973), followed by DT and SVM. All models detected most attacks but misclassified many benign flows: FPR was 20.2% for RF, 21.0% for DT and 37.9% for SVM, and every pairwise difference was significant (McNemar, $p < 0.001$). On Bot-IoT, RF and DT misclassified 5 and 11 of 733,705 flows, a non-significant difference (exact McNemar, $p = 0.070$); the SVM raised no false alarms but missed 2,940 attacks. With only 107 normal test flows, FPR estimates are imprecise: the 95% interval for RF spans 2.0% to 10.5%, and the SVM's zero FPR is compatible with a true rate of up to 3.5%. Pooling the cross-validation folds gave a within-training-partition RF estimate of 2.2% (8 of 370; 95% interval 1.1%–4.2%), compatible with the test interval, although the UNSW-NB15 results below show that such estimates can be optimistic.

Cross-validation and duplicate-free evaluation

Table 5: Five-fold cross-validation on the training partitions

Dataset	**Model**	**F1**	**Balanced accuracy**	**Pooled FPR (false positives)**	**Balanced accuracy, grouped on selected features**	**Balanced accuracy, stratified nested**	**Balanced accuracy, stratified, fixed features**
UNSW-NB15	RF	0.954 ± 0.001	0.932 ± 0.001	8.3% (4,630)	0.914 ± 0.023	0.938	0.938
UNSW-NB15	DT	0.945 ± 0.001	0.922 ± 0.001	9.1% (5,090)	0.915 ± 0.008	0.929	0.929
UNSW-NB15	SVM	0.950 ± 0.003	0.900 ± 0.008	18.0% (1,148)	—	0.901	0.901
Bot-IoT	RF	1.000 ± 0.000	0.989 ± 0.008	2.2% (8)	—	0.992	0.991
Bot-IoT	DT	1.000 ± 0.000	0.976 ± 0.015	4.9% (18)	—	0.986	0.988
Bot-IoT	SVM	0.997 ± 0.001	0.988 ± 0.008	1.9% (7)	—	0.991	0.991

Note: The first three result columns use the reported design (selection and SMOTE refitted in every training fold; identical vectors over all candidate features kept in one fold); mean ± SD over folds. 'Grouped on selected features' keeps records identical on the model inputs in one fold (UNSW-NB15 tree models). The last two columns are less strict designs on the same records.

Cross-validation (Table 5) preserved the model ranking. Refitting feature selection inside each fold changed nothing material: the same 20 features were selected in 5 of 5 folds, and RF balanced accuracy was 0.938 with and 0.938 without refitting. Grouping identical vectors lowered it to 0.932, and grouping on the selected features to 0.914. Duplication on the model inputs therefore accounts for about two fifths of the gap between cross-validated and test balanced accuracy (0.877); the remainder reflects a shift between the official partitions that within-partition validation does not reveal.

Table 6: UNSW-NB15 performance on the full test partition, on test flows absent from training, and after deduplicated re-training

Evaluation set (test records)	Model	Threshold	TPR	FPR [95% CI]	Balanced accuracy	F1
Full test (82,332)	RF	Default	0.956	0.202 [0.198, 0.206]	0.877	0.902
Full test (82,332)	RF	Low-alarm	0.806	0.022 [0.021, 0.024]	0.892	0.884
Full test (82,332)	SVM	Low-alarm	0.199	0.021 [0.020, 0.023]	0.589	0.327
Unseen, selected features (50,548)	RF	Default	0.911	0.196 [0.192, 0.200]	0.857	0.811
Unseen, selected features (50,548)	RF	Low-alarm	0.577	0.020 [0.018, 0.021]	0.778	0.716
Unseen, selected features (50,548)	SVM	Low-alarm	0.377	0.023 [0.022, 0.025]	0.677	0.532
Unseen, all features (73,791)	RF	Low-alarm	0.774	0.023 [0.021, 0.024]	0.876	0.863
Deduplicated training, unseen	RF	Default	0.969	0.271	0.849	0.873
Deduplicated training, unseen	RF	Low-alarm	0.763	0.019	0.872	0.857

Note: Unseen flows are test records whose feature vector does not occur in training. Deduplicated training removed 73,590 duplicated and 940 conflicting records before repeating the full pipeline.

Table 6 reports results on test flows absent from training. On the selected features this subset contains 50,548 flows, only 18,618 of them attacks. At the default threshold RF detected 91.1% of unseen attacks, close to its full-test value, but at the low-alarm threshold detection fell from 80.6% to 57.7%, while FPR stayed near 2.0%. For RF, attacks that remain detectable under a tight budget are disproportionately those duplicated in training; the SVM showed the opposite pattern (19.9% overall, 37.7% unseen). Much of the RF drop reflects a different mix of attack types: Generic attacks, detected at 98.8% at this threshold, made up 41.6% of test attacks but only 4.1% of unseen ones, whereas the poorly detected Fuzzers (26.1%) were over-represented. Detection also fell within some categories, such as DoS (83.6% to 61.3%). Re-training on deduplicated data did not improve generalisation (ROC-AUC 0.972; low-alarm FPR 1.8%, TPR 0.796), so the main conclusions do not depend on the handling of duplicates.

Alarm workload and operating thresholds (RQ3)

Table 7: Effect of the operating threshold on UNSW-NB15 detection, alarm precision and illustrative alarm load

Model	Threshold (FPR budget)	Validation FPR	Test FPR	Test TPR	TPR on unseen flows	PPV at 10%	PPV at 1%	PPV at 0.1%	False alarms per day*
RF	Default	—	20.19%	0.956	0.911	34.50%	4.60%	0.50%	20,195
RF	0.10%	0.00%	no alarms raised						
RF	0.50%	0.50%	1.06%	0.768	0.485	89.00%	42.30%	6.80%	1,057
RF	1.0% (low-alarm)	0.93%	2.21%	0.806	0.577	80.20%	26.90%	3.50%	2,214
RF	2.00%	1.99%	5.18%	0.86	0.704	64.80%	14.40%	1.60%	5,181
RF	5.00%	4.95%	14.00%	0.935	0.873	42.60%	6.30%	0.70%	14,003
SVM	Default	—	37.91%	0.978	0.961	22.30%	2.50%	0.30%	37,911
SVM	0.10%	0.10%	0.28%	0.042	0.099	62.30%	13.10%	1.50%	281
SVM	0.50%	0.50%	1.08%	0.128	0.292	56.90%	10.70%	1.20%	1,076

SVM	1.0% (low-alarm)	1.00%	2.11%	0.199	0.377	51.20%	8.70%	0.90%	2,105
SVM	2.00%	2.00%	4.24%	0.396	0.506	51.00%	8.60%	0.90%	4,238
SVM	5.00%	5.00%	11.18%	0.842	0.641	45.60%	7.10%	0.70%	11,181
DT	Default	—	20.95%	0.926	0.838	32.90%	4.30%	0.40%	20,951

Note: Budget thresholds were chosen on the validation split only. PPV is the share of alarms that are genuine at the stated attack prevalence. *Per 100,000 benign flows (illustrative). For RF a 0.1% budget is unattainable because more than 0.1% of normal validation flows score 1.0; the DT's almost binary scores allow no budget threshold. Alarm precision curves are shown in Figure S4.

Table 7 translates these rates into alarm workload. At the default threshold, RF would raise about 20,195 false alarms per 100,000 benign flows, and at 1% prevalence only 4.6% of its alarms would be genuine. The low-alarm threshold cut test FPR about ninefold to 2.2% while still detecting 80.6% of attacks, raising precision at 1% prevalence to 26.9%, but at 0.1% only 3.5%. Tightening the budget to 0.5% lowered FPR to 1.1% and detected 76.8% of attacks but only 48.5% of unseen ones. At every budget the test FPR was two to three times the validation FPR, so a budget calibrated before deployment would be exceeded after the shift observed here, and no setting reached alarm precision above 6.8% at 0.1% prevalence. For Bot-IoT, alarm precision depends on an FPR that is poorly determined: at 1% prevalence, 17.8% of RF alarms would be genuine at the point estimate but only 8.8% at the upper end of its 95% interval. Inference costs differed by orders of magnitude (Table S2): on UNSW-NB15 the DT needed well under one microsecond per flow and RF 4.5 µs on one thread, whereas the SVM needed about 104 µs.

Explaining the detector (RQ4)

Figure 1 shows permutation importance for the RF models. On UNSW-NB15, shuffling the source time-to-live reduced balanced accuracy by 0.134, more than twice the effect of any other feature, followed by the volume of bytes sent by the source, a connection-state indicator derived from time-to-live values, and the mean size of source packets. On Bot-IoT, the number of inbound connections per destination address and the destination port dominated, consistent with flooding and scanning directed at a few targets. The Mutual Information ranking (Figure S1) agreed closely.

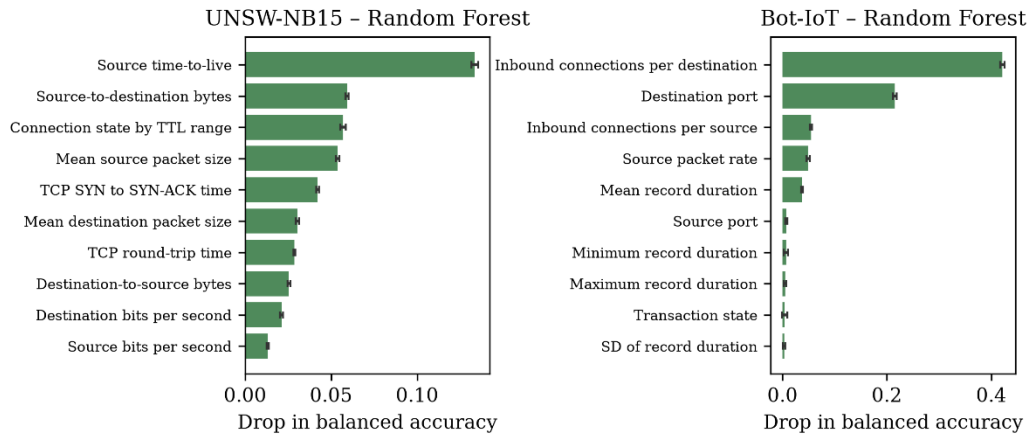

Figure 1: Permutation importance of Random Forest features (mean drop in balanced accuracy ± SD, ten repeats)

DISCUSSION

Benchmark accuracy and operational readiness

Attributes distributed with the datasets were sufficient for a Decision Tree to score perfectly on both benchmarks, so any study that does not remove them explicitly risks reporting leakage rather than detection (Kaufman et al., 2012; Arp et al., 2022). Duplication had a modest effect on test metrics but accounted for about two fifths of the optimism of cross-validation, and for RF it mattered most under a tight alarm budget, where duplicated attacks

were detected far more reliably than others. The larger source of optimism was the difference between the official partitions, consistent with evidence that benchmark-trained detectors transfer poorly (Layeghy & Portmann, 2023). For RQ2, RF was the most accurate model on UNSW-NB15, in line with previous findings (Aljohani et al., 2024; Wakili et al., 2025), but statistically indistinguishable from DT on Bot-IoT. DT offers near-instant inference and transparent rules but almost binary scores, which removes the option of tuning the alarm threshold; the SVM was two orders of magnitude slower and raised the most false alarms. Near-perfect published results on these benchmarks should therefore be read with caution unless leakage controls, duplicate handling and the evaluation partition are reported (Flood et al., 2024).

Alarm workload, explanation and trust

For SMEs and households, the relevant criteria are alarm volume and precision rather than accuracy. At default settings even the best model would flag about one benign flow in five. Threshold selection cut this about ninefold while retaining 80.6% of detections, so the operating point is itself a design decision that weighs missed attacks against analyst time and alarm fatigue (Alahmadi et al., 2022). Because validation-calibrated budgets were exceeded two- to threefold on the test partition, budgets should be monitored and re-calibrated after deployment. With the low attack base rates of typical small networks, alarm precision remained poor at every threshold examined (Axelsson, 2000), so such detectors are better positioned as triage aids than as autonomous blocking systems. For RQ4, a few intuitive features drove the RF decisions and can be expressed in plain language, such as an unusual number of connections to one device. The dominance of time-to-live and destination-specific counts may, however, reflect testbed configuration rather than transferable behaviour (Sommer & Paxson, 2010), so explanations also serve as a diagnostic for spurious correlations before deployment.

Generalisation beyond the benchmarks

A direct cross-dataset test was not possible because the original UNSW-NB15 and Bot-IoT files use different feature sets. NetFlow-based versions with a common feature set exist (Sarhan et al., 2022), and models trained on one of them rarely generalise to another (Layeghy & Portmann, 2023); repeating the present leakage, duplicate and alarm-workload analysis in that setting is a natural next step. Both benchmarks were also generated on testbeds with scripted attacks (Engelen et al., 2021; Flood et al., 2024). Evaluation on labelled traffic from SME or household networks, collected over time so that drift can be measured, is needed to establish realistic false-positive rates; until then, the alarm-precision figures here are likely optimistic.

Contribution relative to prior work

Compared with applied studies reporting near-perfect accuracy (El-Sofany et al., 2024; Pokhrel et al., 2021), this study shows that such results can arise from attributes distributed with the data and that leak-free performance implies a substantial alarm burden. Compared with methodological work (Arp et al., 2022; Apruzzese et al., 2023; Kapoor & Narayanan, 2023), it quantifies several pitfalls at once on two widely used IoT benchmarks, separating leakage, duplication, nested selection and partition shift, and expresses the outcome as alarm precision. It proposes no new detector; its contribution is an evaluation protocol and evidence base. In practice, buyers of IoT security products should ask for false-positive rates and alarm precision at realistic prevalence, vendors should expose the threshold as a configurable budget, and researchers should report the partition, excluded attributes, duplicate handling and threshold procedure.

Limitations

Both datasets come from laboratory testbeds, and no cross-dataset or real-world validation was performed. Only binary detection with three classical models at default settings was examined, chosen because they are widely used and need no tuning expertise; stronger models might perform differently, although the leakage, duplication and base-rate effects arise from the data and evaluation design rather than the model. Confidence intervals do not capture training variation. Bot-IoT's 107 normal test and 370 normal training flows leave its false-positive rates uncertain and could not support threshold selection. The daily traffic volume is illustrative, and no operators were consulted.

CONCLUSION

This study evaluated RF, DT and SVM detectors for IoT botnet traffic on the official UNSW-NB15 and Bot-IoT partitions under explicit leakage controls, with uncertainty quantified and performance translated into alarm workload. With identifier and label-derived attributes retained, a Decision Tree scored perfectly on both benchmarks. Without them, RF remained the strongest model but at default settings raised false alarms on about one fifth of benign UNSW-NB15 flows; a validation-chosen threshold reduced this about ninefold, at a cost in detection borne mainly by attacks not duplicated in training, yet alarm precision would still be low at realistic prevalence. Nested feature selection made no difference, duplication on the model inputs explained about two fifths of the gap between validation and test performance, and the rest reflected a shift between the official partitions. For resource-constrained organisations, the value of IoT botnet detection depends as much on false-alarm control, explanation and evaluation transparency as on accuracy. Future work should repeat the analysis across datasets with common NetFlow features, validate it on real SME and household traffic, and apply the same protocol to multi-class detection and stronger models.

ACKNOWLEDGEMENTS

The authors thank UNITAR International University for providing the academic resources and environment that supported this research. Artificial intelligence (AI) tools were used for editing and proofreading of this paper; the authors reviewed all content and take full responsibility for it.

REFERENCES

1. Alaba, F. A., Othman, M., Hashem, I. A. T., & Alotaibi, F. (2017). Internet of Things security: A survey. *Journal of Network and Computer Applications, 88*, 10–28. https://doi.org/10.1016/j.jnca.2017.04.002
2. Alahmadi, B. A., Axon, L., & Martinovic, I. (2022). 99% false positives: A qualitative study of SOC analysts' perspectives on security alarms. In *Proceedings of the 31st USENIX Security Symposium* (pp. 2783–2800). USENIX Association. https://www.usenix.org/conference/usenixsecurity22/presentation/alahmadi
3. Aljohani, R., Bushnag, A., & Alessa, A. (2024). AI-based intrusion detection for a secure Internet of Things (IoT). *Journal of Network and Systems Management, 32*(3), Article 56. https://doi.org/10.1007/s10922-024-09829-5
4. Antonakakis, M., April, T., Bailey, M., Bernhard, M., Bursztein, E., Cochran, J., Durumeric, Z., Halderman, J. A., Invernizzi, L., Kallitsis, M., Kumar, D., Lever, C., Ma, Z., Mason, J., Menscher, D., Seaman, C., Sullivan, N., Thomas, K., & Zhou, Y. (2017). Understanding the Mirai botnet. In *Proceedings of the 26th USENIX Security Symposium* (pp. 1093–1110). USENIX Association. https://www.usenix.org/conference/usenixsecurity17/technical-sessions/presentation/antonakakis
5. Apruzzese, G., Laskov, P., & Schneider, J. (2023). SoK: Pragmatic assessment of machine learning for network intrusion detection. In *2023 IEEE 8th European Symposium on Security and Privacy (EuroS&P)* (pp. 592–614). IEEE. https://doi.org/10.1109/EuroSP57164.2023.00042
6. Arp, D., Quiring, E., Pendlebury, F., Warnecke, A., Pierazzi, F., Wressnegger, C., Cavallaro, L., & Rieck, K. (2022). Dos and don'ts of machine learning in computer security. In *Proceedings of the 31st USENIX Security Symposium* (pp. 3971–3988). USENIX Association. https://www.usenix.org/conference/usenixsecurity22/presentation/arp
7. Axelsson, S. (2000). The base-rate fallacy and the difficulty of intrusion detection. *ACM Transactions on Information and System Security, 3*(3), 186–205. https://doi.org/10.1145/357830.357849
8. Breiman, L. (2001). Random forests. *Machine Learning, 45*(1), 5–32. https://doi.org/10.1023/A:1010933404324
9. Chawla, N. V., Bowyer, K. W., Hall, L. O., & Kegelmeyer, W. P. (2002). SMOTE: Synthetic minority over-sampling technique. *Journal of Artificial Intelligence Research, 16*, 321–357. https://doi.org/10.1613/jair.953
10. Dietterich, T. G. (1998). Approximate statistical tests for comparing supervised classification learning algorithms. *Neural Computation, 10*(7), 1895–1923. https://doi.org/10.1162/089976698300017197

11. El-Sofany, H., El-Seoud, S. A., Karam, O. H., & Bouallegue, B. (2024). Using machine learning algorithms to enhance IoT system security. *Scientific Reports, 14*(1), Article 12077. https://doi.org/10.1038/s41598-024-62861-y
12. Engelen, G., Rimmer, V., & Joosen, W. (2021). Troubleshooting an intrusion detection dataset: The CICIDS2017 case study. In *2021 IEEE Security and Privacy Workshops (SPW)* (pp. 7–12). IEEE. https://doi.org/10.1109/SPW53761.2021.00009
13. Flood, R., Engelen, G., Aspinall, D., & Desmet, L. (2024). Bad design smells in benchmark NIDS datasets. In *2024 IEEE 9th European Symposium on Security and Privacy (EuroS&P)* (pp. 658–675). IEEE. https://doi.org/10.1109/EuroSP60621.2024.00042
14. Gelgi, M., Guan, Y., Arunachala, S., Samba Siva Rao, M., & Dragoni, N. (2024). Systematic literature review of IoT botnet DDOS attacks and evaluation of detection techniques. *Sensors, 24*(11), Article 3571. https://doi.org/10.3390/s24113571
15. Hussain, F., Hussain, R., Hassan, S. A., & Hossain, E. (2020). Machine learning in IoT security: Current solutions and future challenges. *IEEE Communications Surveys & Tutorials, 22*(3), 1686–1721. https://doi.org/10.1109/COMST.2020.2986444
16. Kapoor, S., & Narayanan, A. (2023). Leakage and the reproducibility crisis in machine-learning-based science. *Patterns, 4*(9), Article 100804. https://doi.org/10.1016/j.patter.2023.100804
17. Kaufman, S., Rosset, S., Perlich, C., & Stitelman, O. (2012). Leakage in data mining: Formulation, detection, and avoidance. *ACM Transactions on Knowledge Discovery from Data, 6*(4), 1–21. https://doi.org/10.1145/2382577.2382579
18. Koroniotis, N., Moustafa, N., Sitnikova, E., & Turnbull, B. (2019). Towards the development of realistic botnet dataset in the Internet of Things for network forensic analytics: Bot-IoT dataset. *Future Generation Computer Systems, 100*, 779–796. https://doi.org/10.1016/j.future.2019.05.041
19. Layeghy, S., & Portmann, M. (2023). Explainable cross-domain evaluation of ML-based network intrusion detection systems. *Computers and Electrical Engineering, 108*, Article 108692. https://doi.org/10.1016/j.compeleceng.2023.108692
20. Lemaître, G., Nogueira, F., & Aridas, C. K. (2017). Imbalanced-learn: A Python toolbox to tackle the curse of imbalanced datasets in machine learning. *Journal of Machine Learning Research, 18*(17), 1–5. https://jmlr.org/papers/v18/16-365.html
21. Liu, S., Fauzi, F., & Kok, V. J. (2025). A systematic literature review: Classifying IoT botnet data features based on its lifecycle. *IEEE Access, 13*, 65040–65065. https://doi.org/10.1109/ACCESS.2025.3559635
22. McNemar, Q. (1947). Note on the sampling error of the difference between correlated proportions or percentages. *Psychometrika, 12*(2), 153–157. https://doi.org/10.1007/BF02295996
23. Moustafa, N., & Slay, J. (2015). UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set). In *2015 Military Communications and Information Systems Conference (MilCIS)* (pp. 1–6). IEEE. https://doi.org/10.1109/MilCIS.2015.7348942
24. Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, É. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research, 12*, 2825–2830. https://jmlr.org/papers/v12/pedregosa11a.html
25. Pokhrel, S., Abbas, R., & Aryal, B. (2021). *IoT security: Botnet detection in IoT using machine learning* (arXiv:2104.02231). arXiv. https://doi.org/10.48550/arXiv.2104.02231
26. Saito, T., & Rehmsmeier, M. (2015). The precision-recall plot is more informative than the ROC plot when evaluating binary classifiers on imbalanced datasets. *PLOS ONE, 10*(3), Article e0118432. https://doi.org/10.1371/journal.pone.0118432
27. Sarhan, M., Layeghy, S., & Portmann, M. (2022). Towards a standard feature set for network intrusion detection system datasets. *Mobile Networks and Applications, 27*(1), 357–370. https://doi.org/10.1007/s11036-021-01843-0
28. Sommer, R., & Paxson, V. (2010). Outside the closed world: On using machine learning for network intrusion detection. In *2010 IEEE Symposium on Security and Privacy* (pp. 305–316). IEEE. https://doi.org/10.1109/SP.2010.25
29. Wakili, A. A., Guni, S., Khan, S. A., Yu, W., & Jung, W. (2025). Advancing machine learning strategies for power consumption-based IoT botnet detection. *Sensors, 25*(24), Article 7553. https://doi.org/10.3390/s25247553

30. Wilson, E. B. (1927). Probable inference, the law of succession, and statistical inference. *Journal of the American Statistical Association, 22*(158), 209–212. https://doi.org/10.1080/01621459.1927.10502953

APPENDIX: SUPPLEMENTARY MATERIAL

Table S1: Leakage ablation: test performance with and without identifier and label-derived attributes

Dataset	Model	Identifier and label-derived attributes	Balanced accuracy	F1	FPR
UNSW-NB15	RF	restored (leaky)	0.972	0.971	0
UNSW-NB15	RF	excluded (this study)	0.877	0.902	0.202
UNSW-NB15	DT	restored (leaky)	1	1	0
UNSW-NB15	DT	excluded (this study)	0.858	0.883	0.21
Bot-IoT	RF	restored (leaky)	1	1	0
Bot-IoT	RF	excluded (this study)	0.977	1	0.047
Bot-IoT	DT	restored (leaky)	1	1	0
Bot-IoT	DT	excluded (this study)	0.972	1	0.056

Table S2: Training cost and single-thread inference latency

Dataset	Model	Training records	Training time (s)	Latency (µs per flow)
UNSW-NB15	RF	202,878	1.99	4.5
UNSW-NB15	DT	202,878	0.91	0.06
UNSW-NB15	SVM	20,000	1.56	104.12
Bot-IoT	RF	2,494,594	32.62	1.19
Bot-IoT	DT	2,494,594	8.08	0.03
Bot-IoT	SVM	20,000	0.12	5.65

Note: Training used eight threads for RF and one thread for DT and SVM; latency was measured on one thread.

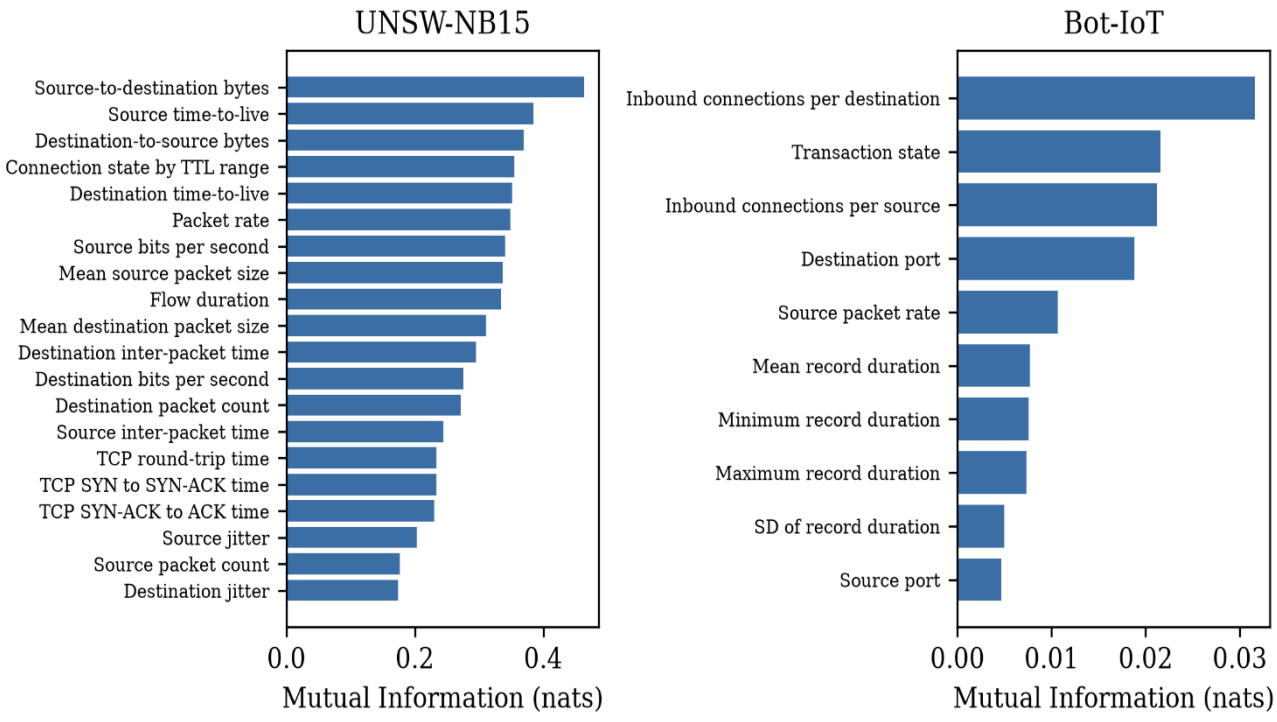

Figure S1: Mutual Information of the retained features (fitting split)

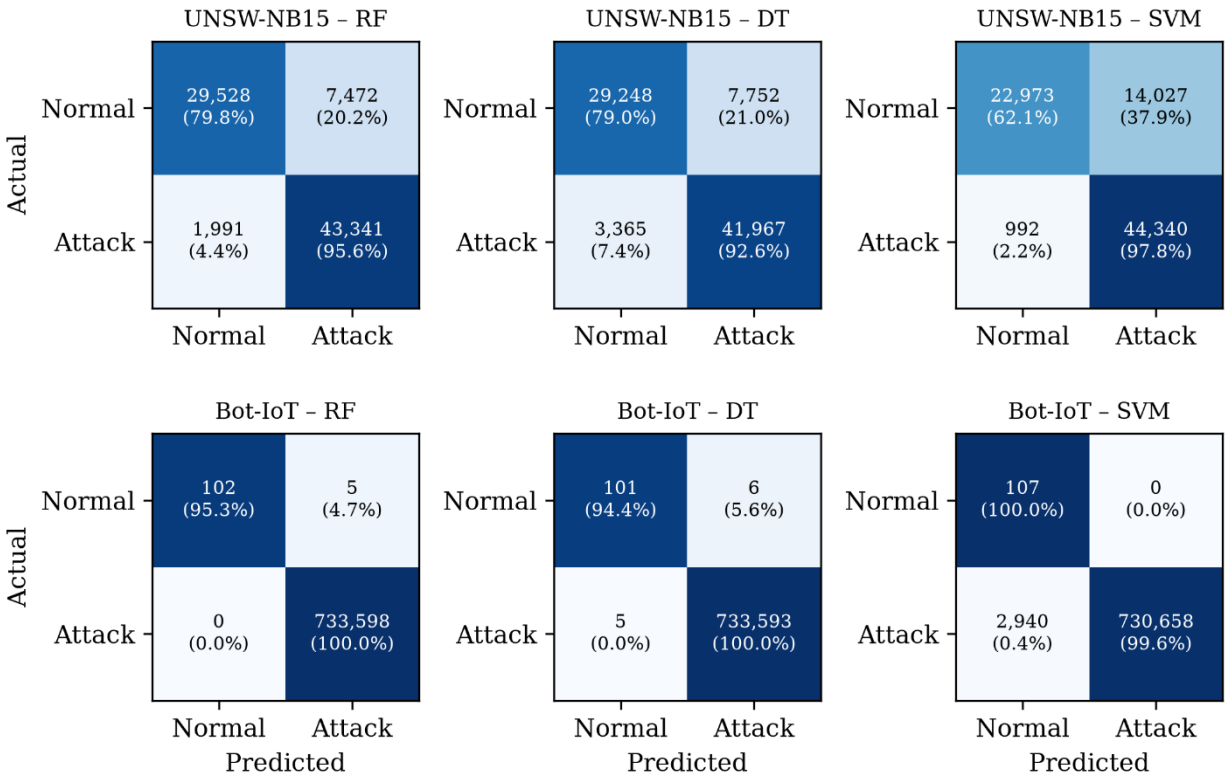

Figure S2: Confusion matrices on the official test partitions (row percentages in brackets)

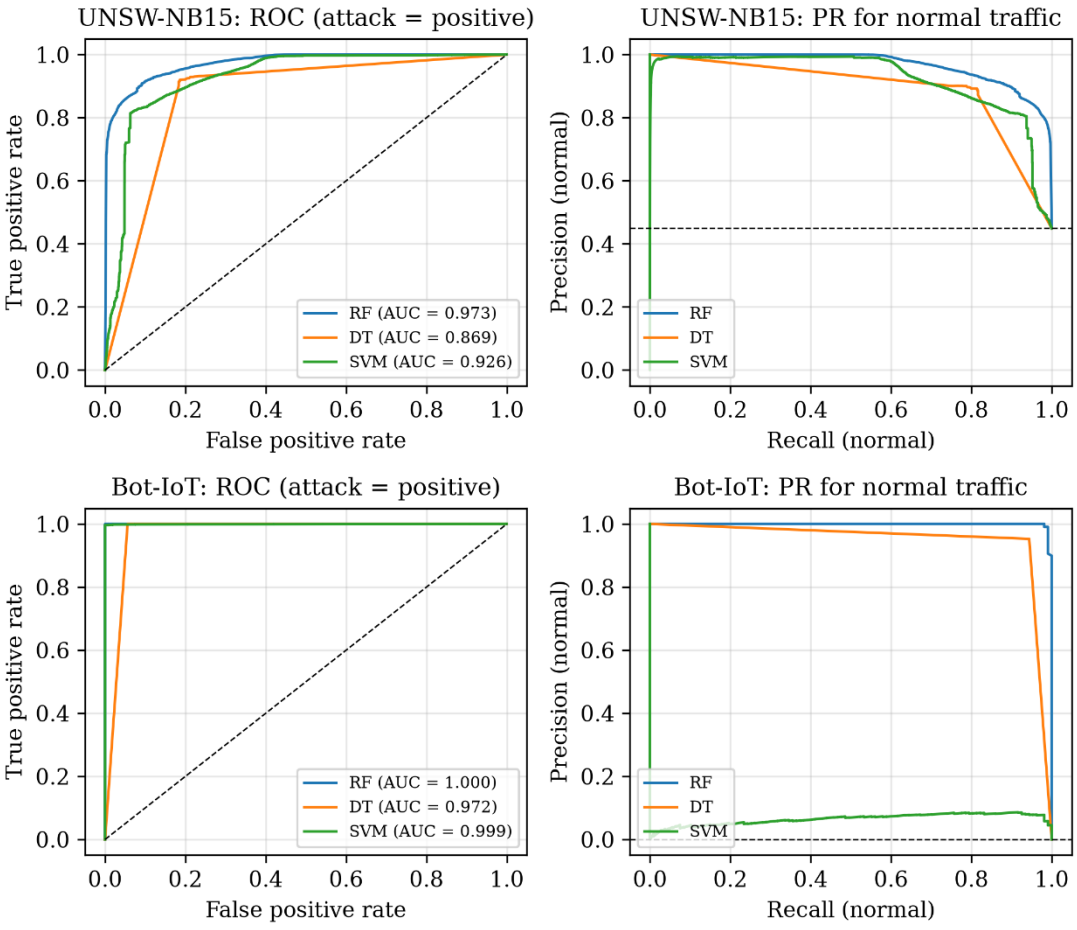

Figure S3: ROC curves (attack as positive class) and precision-recall curves for the normal class

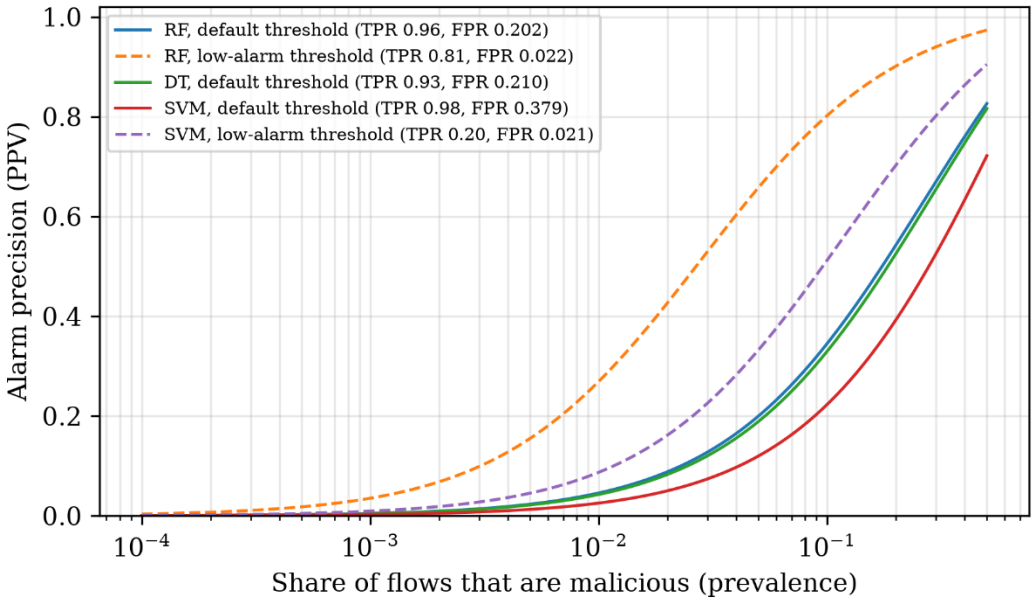

Figure S4: Alarm precision as a function of attack prevalence (UNSW-NB15)