

AN AI-DRIVEN EXPLAINABLE VISUAL ANALYTICS FRAMEWORK FOR AUTOMATED SCIENTIFIC ANALYSIS AND PUBLICATION OF CYBERSECURITY DATA

Gaytri Tanwar¹, Shubham Kumawat², Parveen Kumari³, Mahadeo Prasad⁴, Amit Kumar Rai⁵, Atul Agrawal^{6*}

¹Computer Science & Engineering Affiliation - MAHARSHI DAYANAND UNIVERSITY, Email gaytri.tanwar90@gmail.com , ORCID - 0009-0005-0246-0152

²Department of computer science and Engineering - ITS Engineering College email Shubhamkumawat043@gmail.com, ORCID 0009-0004-8897-5706

³Computer Science and Applications Department, Ganga Institute of Technology and Management, Kablana, Jhajjar Email

:Kumari20parveen@gmail.com, ORCID 0000-0009-9912-8471

⁴Department of Computer Science and Engineering, Arka Jain University Jamshedpur, email: mahadeoprasad@gmail.com ORCID- 0009-0009-3192-9639

⁵Department of computer science and Engineering - ITS Engineering College, greater Noida, U.p email: a.k.raai267@gmail.com , ORCID- 0009-0006-4982-2592

^{6*}Department of computer science and Engineering - ITS Engineering College, greater Noida, U.p email: atulagrawal974@gmail.com orcid: 0009-0004-3601-3919

ABSTRACT

Machine-learning studies of intrusion-detection data often report near-perfect scores from single splits, hand-picked figures and unvalidated explanations. We present XVA-Cyber, a reproducible framework that takes a cybersecurity dataset to statistically validated, explained and provenance-tracked figures and tables. It combines leakage-aware preprocessing, five feature rankers, seven model families, a quantitative SHAP-LIME comparison, Friedman and Holm-corrected Wilcoxon tests with effect sizes, feature-aligned cross-dataset transfer, and ICVRE, a transparent rule-based engine that plans the figures an analysis needs and records why others are omitted. On CIC-IDS2017, CICIOT2023 and UNSW-NB15, de-duplication removed 11.7-36.8% of the records, and label conflicts limited attainable scores. The best multiclass macro-F1 was 0.994, 0.701 and 0.622, respectively, but random forest, XGBoost and LightGBM never differed significantly from one another in any of the six dataset-task runs. TreeSHAP was 22-156 times faster than LIME, and its top features beat a random deletion baseline in 6 of 6 runs against 3 for LIME. Transfer between datasets on aligned features collapsed although within-dataset detectors on the same features performed well. ICVRE's figure choices matched those of published XAI-IDS papers for 80% of the scored recommendations, and each run produced 12 figures and 8 tables from one command. A re-execution reproduced 49 of 61 artifacts byte for byte; the rest differed only in measured timings or by floating-point noise below 10^{-7} .

KEYWORDS: intrusion detection; explainable artificial intelligence; visual analytics; visualization recommendation; statistical validation; reproducibility; cross-dataset generalization

INTRODUCTION

Network intrusion detection produces data that are large, heterogeneous and strongly imbalanced. A single benchmark such as CIC-IDS2017 contains millions of bidirectional flows described by close to eighty statistics, most of which are benign, while some attack classes are represented by a few dozen records [1, 2]. IoT benchmarks add further device and protocol diversity [3], and older but still widely used corpora such as UNSW-NB15 were generated with different tools and feature definitions [4, 5]. Analysing such data well requires careful cleaning, leakage-aware evaluation and metrics that remain informative under imbalance [6, 7].

Machine-learning (ML) detectors routinely report near-perfect accuracy on these benchmarks, yet the same models often fail when the traffic distribution changes [8-10]. Part of the problem lies in the data: labelling and flow-construction errors have been documented in CIC-IDS2017 [2, 11-13], and exact duplicates can inflate test scores when they appear on both sides of a split [7]. Another part lies in how results are reported. Many studies compare mean scores without significance tests, confidence intervals or effect sizes [14, 15], so it is unclear whether a reported improvement exceeds run-to-run variation.

Explainable AI (XAI) methods such as SHAP [16, 17] and LIME [18] are now common in intrusion-detection research [19-21]. They help analysts see which flow statistics drive a decision, but they also add cost and their own uncertainty: explanations can vary between runs, disagree with each other and are rarely evaluated quantitatively [22-24]. In most studies explanations are shown as a handful of static plots chosen by hand.

The step from analysis to publication is also largely manual. Figures and tables are assembled by hand, which makes them hard to regenerate when data or code change, and makes it difficult for readers to link a plotted number back to the run that produced it [25, 26]. Visualization-recommendation systems automate chart choice for general tabular data [27-30], and LLM-based tools now draft charts and even complete papers [31-33], but none of these is tailored to security analytics, where the relevant views depend on class imbalance, model errors and attribution results rather than on column types alone. Reproducibility guidance for ML-based science [34-37] is seldom operationalised inside security pipelines.

This paper addresses these gaps with an integrated framework, XVA-Cyber, that takes a cybersecurity dataset to publication-ready, provenance-tracked figures and tables. The framework chains leakage-aware preprocessing, five feature-ranking methods, a seven-model engine, SHAP and LIME with quantitative comparison, non-parametric statistical validation, cross-dataset evaluation, and a rule-based Intelligent Cybersecurity Visualization Recommendation Engine (ICVRE) that decides which figures a given analysis need. We evaluate it on CIC-IDS2017, CICIoT2023 and UNSW-NB15.

Contributions. (i) An end-to-end, configuration-driven framework that records every figure and table with its caption, data source and SHA-256 hash, and verifies by re-execution which outputs are reproduced byte for byte. (ii) ICVRE, a transparent rule base that profiles both the data and the analysis artifacts, recommends and justifies figures, and reports the views it suppresses. (iii) A measured comparison of SHAP and LIME covering cost, stability, agreement and deletion fidelity. (iv) An evaluation protocol with repeated cross-validation, Friedman and Holm-corrected Wilcoxon tests, bootstrap intervals, a component ablation and a feature-aligned cross-dataset transfer experiment. The individual techniques are established; the novelty lies in their integration, in ICVRE, and in the measured evidence about what each component contributes.

The remainder of the paper is organised as follows. Section 2 reviews related work and Section 3 states the research gap. Sections 4 and 5 describe the framework and methodology, Sections 6 and 7 the datasets and set-up. Sections 8 to 13 report results, and Sections 14 to 16 discuss them, list limitations and conclude.

RELATED WORK

Machine learning for intrusion detection

Tree ensembles remain the strongest general-purpose detectors on flow features. XGBoost [38] and LightGBM [39] dominate recent benchmarks on CIC-IDS2017, UNSW-NB15 and CICIoT2023 [40-42], while deep models such as CNN-LSTM hybrids report comparable scores on IoT data [43-45]. Almahaqeri et al. [41] evaluated gradient boosting on CICIoT2023 in binary, 8-class and 34-class settings and showed through ablation that halving the feature set changed accuracy only marginally while cutting inference latency. Such single-dataset results rarely come with significance tests; Elangovan et al. [10] are an exception, using Wilcoxon tests to compare models across four datasets.

Dataset quality and generalisation

Several audits of CIC-IDS2017 found incorrectly constructed flows and mislabelled attacks; Engelen et al. [2] reconstructed or relabelled more than a fifth of the traces, and Liu et al. [11] reported further errors in both CIC-IDS2017 and CSE-CIC-IDS2018. Rosay et al. [13] and Lanvin et al. [12] showed that corrections change detection results. Generalisation across datasets is weak: D'hooge et al. [8] found that models trained on one CIC dataset failed on the other, and Cantone et al. [9] observed cross-dataset accuracy close to chance for most attack-dataset combinations. Sarhan et al. [46, 47] argue that incompatible feature sets are one cause and propose NetFlow-based standard features. Kılıç and Şengül [48] showed that SHAP-guided feature subsets improve CIC-IDS2017-to-CSE-CIC-IDS2018 transfer only slightly. Pitfalls such as sampling bias, data snooping and inappropriate baselines are widespread in security ML [6], and leakage has affected hundreds of ML-based papers across disciplines [7].

Explainable intrusion detection

Surveys by Neupane et al. [19], Capuano et al. [20], Moustafa et al. [21], Mohale and Obagbuwa [49] and Al and Sagiroglu [50] map XAI methods onto IDS tasks and repeatedly call for standard ways to evaluate explanations. Applied work mostly adds SHAP or LIME to a detector: Patil et al. [51] used LIME with a voting ensemble on CIC-IDS2017, Larriva-Novo et al. [52, 53] integrated SHAP into a real-time IDS and into reinforcement-learning-based categorisation on UNSW-NB15, and Mohale and Obagbuwa [54] compared LIME, SHAP and ELI5 across seven models. SHAP has also been used for feature selection [40, 55]. Fewer studies evaluate the explanations themselves. Arreche et al. proposed an end-to-end XAI framework over three datasets [56] and an evaluation suite with six explanation metrics [22]; Gaspar et al. [57] compared LIME and SHAP by perturbation analysis and a user survey; Hermosilla et al. [23] assessed fidelity and stability of SHAP and LIME on UNSW-NB15; and Nugraha et al. [24] cross-checked the two methods and measured explanation time. Muhammad et al. [58] combined LIME and ELI5 into a global-local framework.

Visual analytics for cybersecurity

Visualization has a long history in network security [59]. Systems such as Situ [60] couple streaming anomaly detection with interactive views deployed in a security operations centre, and recent work visualises attack graphs from honeynets [61], inter-domain routing threats [62] and flow records [63]. Chavan and Pappas [64] combined deep vulnerability detection with SHAP-based and CVSS-aligned plots as an analyst layer. Hohman et al. [65] survey visual analytics for understanding deep models. These systems target interactive exploration; they do not decide which figures a scientific report of an ML study should contain, and they are not linked to statistical validation.

AI-driven and automated visualization

Visualization recommendation automates chart selection for tabular data. Draco encodes design knowledge as constraints [28]; VizML [27] and Qian et al. [66] learn design choices from large corpora; KG4Vis [29] and AdaVis [30] add explainable, knowledge-graph-based recommendations; MultiVision [67] recommends multi-view dashboards; and Zhu et al. [68] survey the field. LLM-based systems such as LIDA [31] and LLM4Vis [32] generate and justify charts from natural language. All of these reason mainly about column types and statistical properties of a single table. Security-ML

reporting needs views of model outputs, errors, attributions and test results as well, and needs them to be evidence-gated rather than exhaustive.

Reproducible computing and AI-assisted publishing

Community guidance for reproducible ML asks for code and data release, fixed seeds, documented environments and model info sheets [34-37], consistent with earlier rules for computational research [25] and FAIR data principles [69]. Deployment studies show how pipelines break when these practices are missing [70]. Tools such as pylustrator [71] make individual figures reproducible, and Rougier et al. [26] give design rules for scientific figures. At the other end of the spectrum, LLM pipelines now draft full papers [33], and LLM-assisted writing is spreading quickly in the literature [72, 73]. This raises concerns about fabricated content and references [74, 75]. Our framework automates the evidence side of publishing (figures, tables, captions and provenance) and leaves claims and prose to the authors.

Research Gap

Table 1 compares nineteen studies with the proposed framework on nine capabilities. Marks were assigned from evidence in each paper: full text where it was accessible, and the abstract otherwise (marked †). A ✓ means the capability was implemented and reported, ○ that it was partly addressed, and - that we found no evidence of it.

Table 1. Comparison of recent studies with the proposed framework (✓ implemented, ○ partially addressed, - no evidence found; † assessed from the abstract only).

Paper	AI/ML	XAI	Visual Analytics	Multi-Dataset	Statistical Testing	Auto Visualization	Auto Tables	Publication Automation	Reproducibility
Chavan & Pappas (2025)	✓	✓	○	✓	✓	-	-	-	○
Larriva-Novo et al. (2023)	✓	✓	○	-	-	-	-	-	-
Arreche et al. (2024) XAI-IDS	✓	✓	○	✓	-	-	-	-	○
Larriva-Novo et al. (2024)	✓	✓	-	-	-	-	-	-	-
Nugraha et al. (2025)	✓	✓	-	✓	-	-	-	-	○
Ahmed et al. (2024)	✓	✓	-	-	-	-	-	-	-
Chen et al. (2024)	✓	✓	-	✓	-	-	-	-	-
Hermosilla et al. (2025)	✓	✓	-	-	-	-	-	-	○
Almahaqeri et al. (2026)	✓	-	-	-	-	-	-	-	○
Yaras & Dener (2024)	✓	-	-	✓	-	-	-	-	○
Elangovan et al. (2026)	✓	-	-	✓	✓	-	-	-	○
Patil et al. (2022)	✓	✓	-	-	-	-	-	-	○
Gaspar et al. (2024)†	✓	✓	-	-	-	-	-	-	-

Arreche et al. (2024) E-XAI†	✓	✓	-	✓	-	-	-	-	○
Cantone et al. (2024)†	✓	-	○	✓	-	-	-	-	-
Kılıç & Şengül (2026)†	✓	✓	-	✓	-	-	-	-	-
Mohale & Obagbuwa (2025)†	✓	✓	-	-	-	-	-	-	-
Goodall et al. (2019) Situ†	✓	○	✓	-	-	-	-	-	-
Rabzelj et al. (2023)†	○	-	✓	-	-	-	-	-	-
Proposed framework (this work)	✓	✓	✓	✓	✓	✓	✓	✓	✓

Five gaps follow from the table and the review above.

G1-Validation without statistics. Only two of the nineteen studies test the significance of model differences [10, 64]. Most compare means on one split, so reported improvements cannot be separated from noise [14, 15].

G2-Explanations without evaluation. Thirteen of the nineteen studies use SHAP or LIME, but only a few measure explanation cost, stability or fidelity [22-24, 57], and none links these measurements to feature selection and ablation in one pipeline.

G3-Generalisation claims on one dataset. Ten of the nineteen studies evaluate a single dataset. Where several are used, transfer across incompatible feature spaces is rarely separated from independent re-evaluation [9, 48].

G4-No evidence-driven figure selection for security ML. General visualization recommenders [27-30] do not consider model outputs, errors or attributions, and security visual-analytics systems [60, 61] target exploration rather than reporting. None of the compared studies automates figure or table generation.

G5-Reproducibility as a statement, not a mechanism. Where reproducibility is addressed, it is through code release or reported seeds; none of the studies hashes its outputs or re-executes the pipeline to verify them [35, 36].

Proposed Framework

XVA-Cyber is organised in seven layers (Figure 1). Each layer reads the artifacts of the layer above and writes its own artifacts to disk, so any stage can be inspected or re-run. A single YAML configuration file fixes every setting, and a run tracker stores a snapshot of it together with the random seed, package versions, hardware description, stage timings, a structured log and the SHA-256 hash of every result, figure and table.

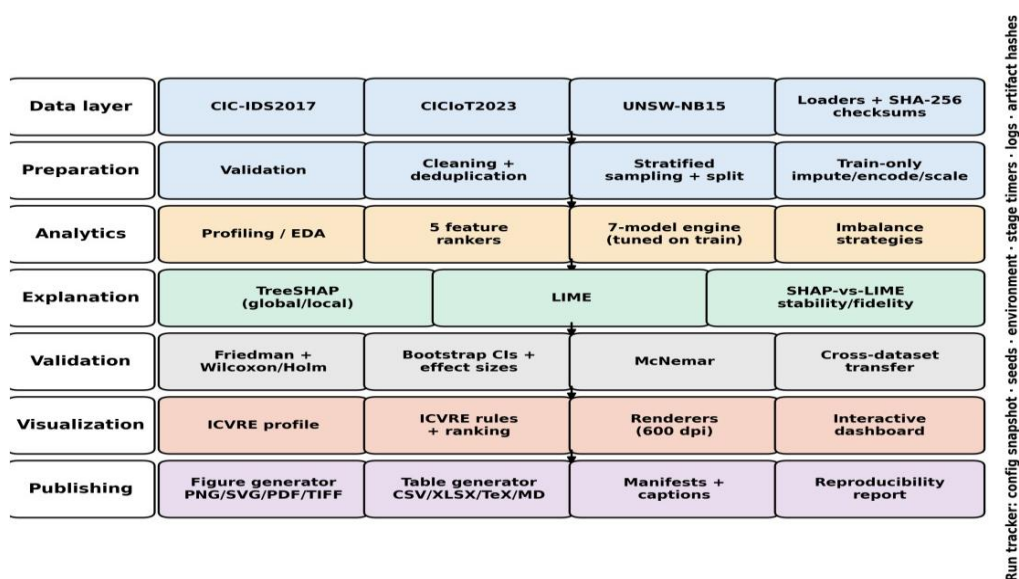


Figure 1. Architecture of the proposed framework. Each layer consumes the artifacts of the layer above; a run tracker records configuration, seeds, environment, stage timings and SHA-256 hashes of every output.

Data layer. Dataset-specific loaders read the original CSV files, record each file's size, row count and SHA-256 checksum, drop identifier columns (flow IDs, IP addresses, timestamps where present) and map raw labels to attack families using explicit, version-controlled rules. Unmapped labels raise an error rather than being silently dropped.

Preparation layer. Validation, cleaning, de-duplication, sampling and splitting follow the fixed order of Figure 2. The layer reports every quantity it changes: infinite values, removed duplicates, feature vectors that carry conflicting labels, merged rare classes and per-class sampling counts.

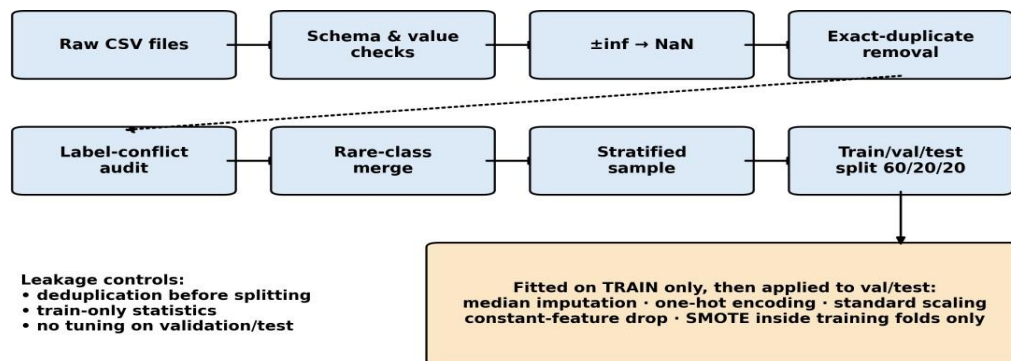


Figure 2. Data-processing workflow. Duplicates are removed before splitting so that identical flows cannot fall in both training and test data; every fitted statistic is estimated from the training split only.

Analytics layer. Five feature rankers (ANOVA F, mutual information, random-forest impurity importance, XGBoost gain and mean |SHAP|) are computed on the training split. The model engine contains one representative of each methodological family: logistic regression (linear), decision tree, random forest (bagging), XGBoost and LightGBM (boosting), k-nearest neighbours (instance-based) and a multilayer perceptron (neural). An RBF-kernel SVM was excluded because its quadratic kernel cost is impractical at this data scale on the test machine and the linear family is already represented.

Explanation layer. TreeSHAP [17] gives exact Shapley values for tree ensembles; LIME [18] fits local surrogate models on perturbed samples. Both are applied to the same stratified set of test flows and compared on the measures of Section 5.4.

Validation layer. Repeated cross-validation, non-parametric tests, bootstrap intervals, McNemar's test and the cross-dataset transfer experiment (Section 5.5).

Visualization layer. ICVRE (Section 5.6; Figure 3) decides which figures the analysis needs, and dedicated renderers draw them. Captions are assembled from the plotted values, so a caption cannot disagree with its figure. A Streamlit dashboard provides interactive views of the same saved results (data profile, model comparison with confidence intervals, confusion matrices, SHAP views, statistical tests, the ICVRE plan and provenance).

Publishing layer. A figure generator applies one style object (600 dpi, bold black text, fixed column widths, colour-blind-safe palette) and exports PNG, SVG, PDF and LZW-compressed TIFF with creation timestamps removed, so identical inputs give byte-identical files. A table generator exports CSV, XLSX, LaTeX, Markdown and a JSON payload used to build the manuscript. Both write manifests listing caption, purpose, data source and file hashes. Figures are never edited by hand.

METHODOLOGY

Preprocessing and leakage control

Let $D = \{(x_i, y_i)\}_{i=1}^N$ be the loaded flows. Positive and negative infinities are replaced by missing values. Exact duplicates of (x_i, y_i) are removed before any split, because a duplicated flow that lands in both training and test sets inflates test scores [6, 7]. To fit the 7 GB test machine, floating-point features are cast to single precision (float32) when loaded, so duplicates are exact at float32 precision; rows that differ only beyond about seven significant digits are treated as duplicates. Section 6 reports how much this changes the counts. We also count feature vectors that occur with more than one label; such conflicts bound the achievable accuracy and are reported rather than removed. Attack families with fewer than 50 flows after de-duplication are merged into an Other-rare class so that stratified 10-fold cross-validation remains possible.

Because the full datasets exceed the memory and time budget of the test machine, a modelling sample of $n = 120,000$ flows is drawn per dataset by proportional stratified sampling with minority retention: class c with N_c flows contributes $nc = \min\{N_c, \max(\text{round}(fN_c), \min(N_c, 1000))\}$ flows, where $f = n/N$. This keeps small attack classes usable at the cost of mildly over-representing them; the exact per-class counts are reported (Table 3). The sample is split 60/20/20 into training, validation and test sets, stratified on the attack family.

All fitted transformations (median imputation, one-hot encoding of categorical fields with a 0.5% minimum-frequency rule, standardisation and removal of constant columns) are estimated on the training split only and then applied unchanged to validation and test data. Three imbalance strategies are compared on the validation split with a LightGBM reference model: none, balanced class weights ($wc = N_{\text{train}}/(K N_c, \text{train})$) and SMOTE [76]. SMOTE is applied only to training data (inside training folds during cross-validation), never before splitting.

Feature selection

Each ranker assigns a score $sm(j)$ to feature j . For the SHAP ranker, $s(j) = 1/|S| \sum_{i \in S} \sum_k |\phi_{ijk}|$ over a training subsample S of 3,000 flows and all classes k . For every ranker and $k \in \{10, 20\}$ the top- k subset is evaluated on the validation split. The chosen subset is the smallest one whose validation macro-F1 is within 0.002 of the best subset (a parsimony rule fixed before the experiments). Agreement between rankers is measured by Kendall's τ over full rankings and by the Jaccard index $J(A, B) = |A \cap B|/|A \cup B|$ of top-10 sets.

Model training and evaluation

Hyperparameters are tuned by random search (6 candidates, 3-fold stratified CV, macro-F1) on a 20,000-flow subsample of the training split. Search spaces are listed in the code and the chosen values in the supplementary tables. Tuned models are refitted on the full training split and evaluated once on the test split. We report accuracy, macro precision, recall and F1, Matthew's correlation coefficient (MCC), one-vs-rest macro ROC-AUC and PR-AUC (average precision), and, for binary tasks, specificity, FPR and FNR. For multiclass tasks, operational attack-vs-benign rates are derived from the multiclass predictions: a benign flow predicted as any attack is a false positive and an attack predicted as benign a false negative. Training time, inference time per flow, serialised model size and peak Python allocation are recorded. Because accuracy is dominated by the majority class, conclusions rest on macro-F1, MCC and PR-AUC.

Explainability and its evaluation

For a model f and instance x , SHAP decomposes the output into $f(x) = 0 + \sum_j \phi_j$ [16]; LIME fits a sparse linear surrogate g minimising $\mathcal{A}(f, g, \pi x) + \Omega(g)$ on perturbed neighbours weighted by πx [18]. SHAP is computed for the boosting model with the higher CV macro-F1 on a stratified set of up to 2,000 test flows (fewer when small classes cannot fill their share). On 40 test flows sampled evenly across classes we measure:

Cost: wall-clock time per explanation.

Agreement: $J(\text{top5SHAP}, \text{top5LIME})$ for the predicted class.

Stability: mean pairwise Jaccard of LIME top-5 sets over five random seeds. TreeSHAP is deterministic for a fixed model, so its stability is measured as the mean pairwise Jaccard of global top-5 sets across three models retrained with different seeds.

Deletion fidelity: the drop $\Delta p = p\hat{y}(x) - p\hat{y}(x')$, where x' replaces the top-5 features by their training medians. The same drop for five random features serves as a baseline, and paired Wilcoxon tests compare each method with it.

These measures describe how well explanations reflect the model; they do not validate the explanations as causal accounts of attacks.

Statistical validation and cross-dataset evaluation

Each tuned model is re-trained in 10-fold stratified CV on a 30,000-flow subsample of the development data (training plus validation), using identical folds for all models. The Friedman test [77] checks whether the seven models differ in macro-F1; Kendall's $W = 12Sn^2(k^3 - k)$ quantifies the effect. Pairwise differences are tested with the Wilcoxon signed-rank test [78], as recommended over mean-rank post-hoc tests [14, 15], with Holm correction across the 21 pairs [79]. The effect size is the matched-pairs rank-biserial correlation $rrb = (R + -R-)/(R + +R-)$. For the two best models, a 10,000-sample bootstrap gives a CI for the mean CV difference, and McNemar's exact test compares their errors on the held-out test set. Test-set metrics carry 95% percentile bootstrap intervals (1,000 resamples; 300 for PR-AUC). Because CV folds share training data, fold scores are not independent and the resulting p-values are optimistic [14]; we therefore interpret them together with effect sizes.

Cross-dataset validation comprises four experiments. A, C and D apply the complete framework independently to CIC-IDS2017, CICIoT2023 and UNSW-NB15. B tests transfer. The three datasets use different feature extractors, so a model trained on the full CIC-IDS2017 feature space cannot be applied to the others. Transfer is therefore evaluated only on features with matching definitions after unit harmonisation (for example flow duration in seconds): seven features for CIC-IDS2017 ↔ UNSW-NB15 (moderate alignment) and five packet-length and rate features for CIC-IDS2017 ↔ CICIoT2023 (weak alignment, because CICIoT2023 aggregates over packet windows rather than complete flows). Features are log-transformed, and each transfer score is paired with a within-target model trained on the same aligned features, so any drop reflects distribution shift rather than lost features. The binary task is used, because it is the only label space shared by all three datasets.

Intelligent Cybersecurity Visualization Recommendation Engine (ICVRE)

ICVRE (Figure 3) builds two profiles. The data profile records the number of rows and features, numeric and categorical counts, presence of a temporal attribute, number of classes, imbalance ratio $\rho = \max_c N_c / \min_c N_c$, missing and infinite cells, the fraction of numeric feature pairs with $|r| > 0.9$, the median absolute skewness and the largest IQR-outlier fraction. The artifact profile records how many models were evaluated, whether probabilities exist, how many feature rankings were computed, whether SHAP and LIME results, statistical tests, multiple datasets and ablation results are available, and the top-ranked features.

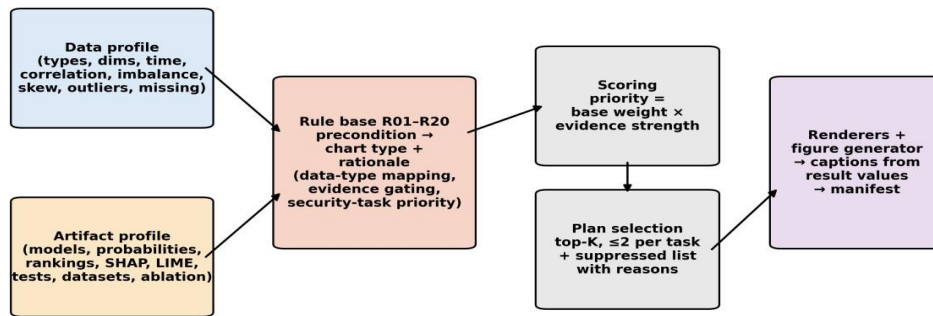


Figure 3. Architecture of the Intelligent Cybersecurity Visualization Recommendation Engine (ICVRE). Profiles of the data and of the analysis artifacts trigger transparent rules; fired rules are scored, filtered by a diversity constraint and rendered, while non-fired rules are reported with the reason for suppression.

Twenty rules (R01-R20) map profile conditions to chart types. They follow three principles. (i) Data-type mapping: numerical $\times$ numerical relations give scatter plots, categorical $\times$ numerical comparisons give box plots, single categorical distributions give bar charts, many numerical variables give heatmaps or projections, and temporal attributes give timelines. (ii) Evidence gating: a chart is proposed only when the profile shows the property it is meant to reveal. A missing-value chart requires missing values, a logarithmic axis requires $\rho > 100$, a redundancy heatmap requires more than 1% of pairs with $|r| > 0.9$, and projections require more than ten dimensions. (iii) Security-task priority: base weights reflect analyst tasks, with detection quality and error structure ranked above descriptive views. Each fired rule receives priority $\pi = w_{rule} \cdot e$, where $e \in [0,1]$ is an evidence strength. For example, $e = 0.5 + 0.5\min(1, \log_{10}\rho/4)$ for the class-balance chart, and the PR curve is up-weighted and the ROC curve down-weighted as ρ grows. Rules are ranked by π , at most two figures per analytic task are kept (a diversity constraint), and the plan is capped at twelve figures. Every non-fired or dropped rule is written to the plan with its reason, so the plan documents why a figure is absent.

ICVRE's weights are design parameters, not learned values. We evaluate the plans against a literature-derived reference. From the figure captions of the seven full-text XAI-IDS papers in our corpus, we coded 44 figures by analytic task and chart family. A planned figure agrees when its chart family appears in the reference set for its task, and coverage is the share of reference tasks for which ICVRE planned a figure. The coding was done by one annotator from captions, and published choices are not a gold standard. We therefore treat these numbers as a consistency check, not as recommendation accuracy. A protocol with independent expert raters is provided but was not executed.

Ablation and publication-workflow measurement

The component ablation compares four configurations built from the same run: A, models on all features; B, models on the selected features; C, B plus the XAI engine; and D, the complete framework (C plus statistical validation, ICVRE figures, tables and provenance). Pipeline time sums preprocessing, feature ranking (B-D), training and inference of all seven models, and the XAI, statistics, figure and table stages where included. Hyperparameter search is identical for all configurations and is excluded. C and D reuse B's models, so their predictive scores are identical by construction; their contribution is measured in outputs (explanations, tests, figures, tables, hashes) and time.

For the publication workflow we measure, for every run of the automated workflow, the number of commands issued, the wall-clock time, the number of generated figures and tables, their generation time, the number of hashed artifacts and the number of manual edits. The conventional manual workflow would need a timed study with human analysts, which we did not conduct. We therefore report no time savings. Reproducibility is tested by re-executing one complete run with the same configuration and comparing all numeric results (excluding timing columns), tables and figure files byte for byte.

Dataset Description

All three datasets were downloaded from their official providers, and every statistic below was measured by the framework from the files actually read (Table 2). SHA-256 checksums of all files are listed in the reproducibility report; the CIC-IDS2017 archive also matched the provider's MD5 checksum. CIC-IDS2017 [1] was captured over five working days in a testbed with profiled benign users and scheduled attacks. We used the eight daily Machine Learning CVE CSV files (2,830,743 flows, 77 features after dropping one duplicated column). The files contained 4,376 infinite and 1,358 missing values, and 330,995 exact duplicates (11.7%). Fifteen raw labels were mapped to nine families; Infiltration and Heartbleed had fewer than 50 flows after de-duplication and were merged into Other-rare.

CICIoT2023 [3] records 33 attacks against a testbed of 105 IoT devices, grouped into seven families plus benign traffic. The provider distributes the CSV release as 63 merged files; because of the memory limit of the test machine, we used the first three (2,158,185 rows, 39 features). Results for CICIoT2023 therefore describe this subset only.

UNSW-NB15 [4, 5] was generated with the IXIA Perfect Storm tool and contains nine attack families. We used the official training and testing partitions (257,673 records, 42 features including three categorical fields). To treat all datasets identically, the two partitions were pooled, de-duplicated and re-split with the same stratified protocol. De-duplication removed 94,928 records (36.8%), and 11,506 of the remaining records share an identical feature vector with a record of another class.

Table 2. Characteristics of the datasets as measured from the downloaded files.

Property	CIC-IDS2017	CICIoT2023	UNSW-NB15
Files read	8	3	2
Raw flows	2830743	2158185	257673
Raw features	77	39	42
Missing cells	1358	52	0
Infinite cells	4376	44	0
Duplicates removed	330995	737714	94928
Duplicate %	11.6929	34.1821	36.8405
Flows after dedup	2499748	1420471	162745
Label-conflict rows	1438	132347	11506
Raw labels	15	34	10
Families	8	8	10
Attack %	17.0368	96.4360	47.3274
Imbalance ratio	44124.8936	1527.4743	501.2982
Modelling sample	122409	121457	120044
Transformed features	69	39	58

Numerical precision of de-duplication. The duplicate and conflict count above are exact at the float32 precision used by the pipeline. An independent recount at the original float64 precision found 308,381 duplicates and 1,396 conflict rows in CIC-IDS2017, 735,814 and 131,173 in CICIoT2023, and identical counts for UNSW-NB15. The difference (at most 0.8% of the CIC-IDS2017 flows) comes from rows that differ only in low-order digits of floating-point features. Such rows are near-duplicates, so merging them does not create leakage, but the counts should be read as float32 counts.

Table 3 lists the class distribution after de-duplication and the number of flows drawn for the modelling sample. Figure 4 shows the imbalance: the ratio between the largest and smallest class is 44,125 in CIC-IDS2017, 1,527 in the CICIoT2023 subset and 501 in UNSW-NB15.

Table 3. Class (attack-family) distribution after duplicate removal and number of flows in the modelling sample.

Dataset	Class	Flows after dedup	Share %	Sampled
CIC-IDS2017	Benign	2073870	82.9632	99556
CIC-IDS2017	DoS	193748	7.7507	9301
CIC-IDS2017	DDoS	128016	5.1212	6145
CIC-IDS2017	PortScan	90819	3.6331	4360
CIC-IDS2017	BruteForce	9152	0.3661	1000
CIC-IDS2017	WebAttack	2143	0.0857	1000
CIC-IDS2017	Botnet	1953	0.0781	1000
CIC-IDS2017	Other-rare	47	0.0019	47
CICIoT2023	DDoS	921067	64.8424	77811
CICIoT2023	DoS	273793	19.2748	23130
CICIoT2023	Mirai	119984	8.4468	10136
CICIoT2023	Benign	50626	3.5640	4277
CICIoT2023	Recon	31745	2.2348	2682
CICIoT2023	Spoofing	21517	1.5148	1818
CICIoT2023	Web	1136	0.0800	1000
CICIoT2023	BruteForce	603	0.0425	603
UNSW-NB15	Benign	85722	52.6726	63207
UNSW-NB15	Exploits	27434	16.8570	20228
UNSW-NB15	Fuzzers	20960	12.8790	15455
UNSW-NB15	Reconnaissance	9991	6.1391	7367

UNSW-NB15	Generic	7599	4.6693	5603
UNSW-NB15	DoS	5500	3.3795	4055
UNSW-NB15	Analysis	2032	1.2486	1498
UNSW-NB15	Backdoor	1880	1.1552	1386
UNSW-NB15	Shellcode	1456	0.8947	1074
UNSW-NB15	Worms	171	0.1051	171

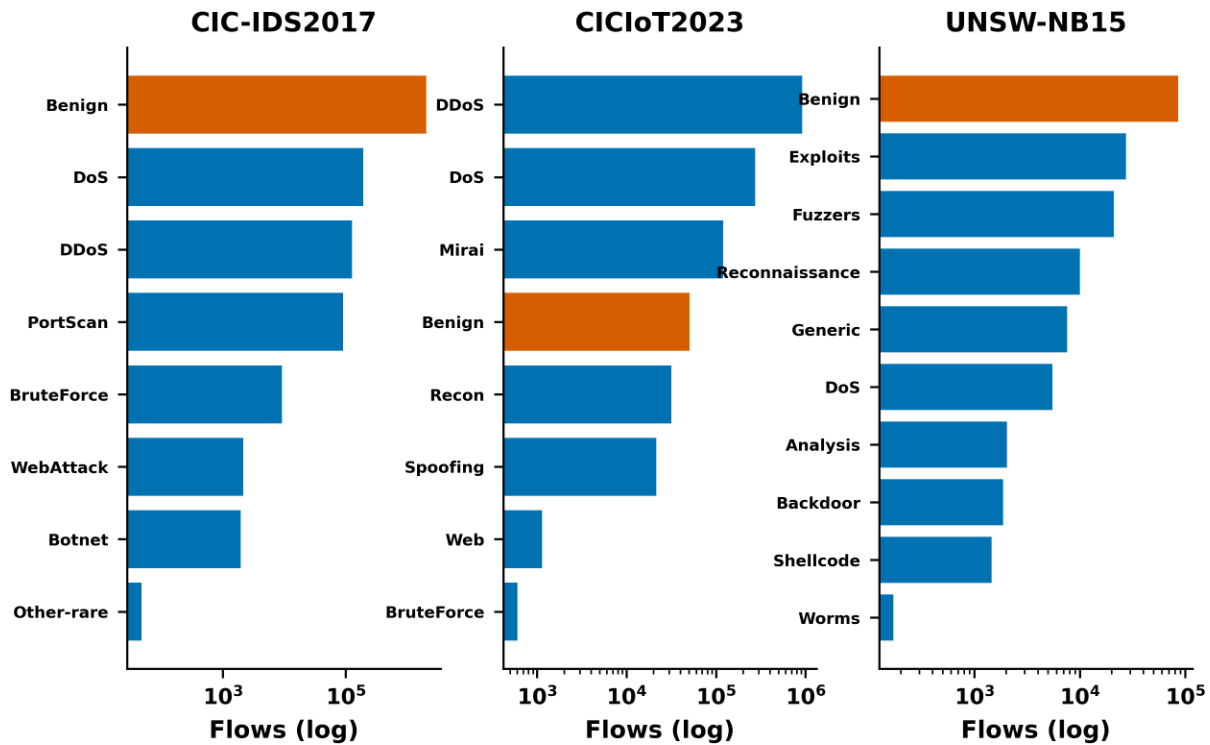


Figure 4. Class distribution after duplicate removal for the three datasets (logarithmic axis; benign traffic in orange). Imbalance ratios are listed in Table 2.

Experimental Setup

All experiments ran on a cloud virtual machine with two vCPUs (Intel Xeon, 2.8 GHz), 7 GB RAM and no GPU, under Linux with Python 3.11. The main libraries were scikit-learn 1.8.0 [80], XGBoost 3.2.0, LightGBM 4.7.0, SHAP 0.51.0, LIME 0.2.0.1, umap-learn 0.5.12 [81], SciPy 1.17.1, statsmodels 0.15.0 and Matplotlib 3.10.9 [82]. The complete environment is recorded with each run. The global random seed was 42 for Python, NumPy and all estimators.

The settings follow Section 5: a 120,000-flow modelling sample per dataset, a 60/20/20 split, random-search tuning with six candidates, and 10-fold CV on 30,000 development flows for the statistical tests. Two settings were changed after a pilot run, and both changes were applied to all datasets before the reported runs. First, LightGBM with library defaults diverged on the CIC-IDS2017 sample (validation macro-F1 0.1541, against 0.9933 with *min_child_samples* = 50 and *lambda* = 1), so the regularised setting became the base configuration. Second, the XAI comparison used top-5 rather than top-10 features, because the smallest selected subset had only ten features, which would have made all top-10 sets identical. Tuned hyperparameters for every model and dataset are exported by the table generator (supplementary table S5).

RESULTS

Data quality

De-duplication had a large effect on every dataset (Table 2). It removed 11.7% of CIC-IDS2017 flows, 36.8% of UNSW-NB15 records and 34.2% of the CICIoT2023 rows. Label conflicts, meaning identical feature vectors carrying different labels, were rare in CIC-IDS2017 (1,438 rows) but frequent in UNSW-NB15 (11,506) and in CICIoT2023 (132,347 rows). Such rows cannot be separated by any model that uses these features, so they bound the attainable scores. Outliers were reported but not removed; ICVRE ranked the outlier chart below its plan cut-off in all runs (supplementary table S6).

Feature selection

The five rankers agreed only partly. On CIC-IDS2017 (Figure 5), no two methods shared more than 0.43 of their top-10 features (Jaccard index). The full rankings of XGBoost gain and SHAP were the most similar (Kendall's $\tau = 0.66$), followed by mutual information and RF impurity ($\tau = 0.64$). The univariate ANOVA-F ranking agreed least with the others ($\tau \leq 0.33$). Different rankers therefore propose different subsets, which is why the subset is chosen on validation

data rather than by a fixed method. The parsimony rule selected 10 features ranked by SHAP for CIC-IDS2017, 10 (XGBoost gain) for UNSW-NB15 and 20 (XGBoost gain) for CICIoT2023 in the multiclass task (Table 4). On the validation split, the imbalance comparison favoured no re-balancing for CIC-IDS2017, no re-balancing (multiclass) and class weighting (binary) for UNSW-NB15, and SMOTE (multiclass) and no re-balancing (binary) for CICIoT2023.

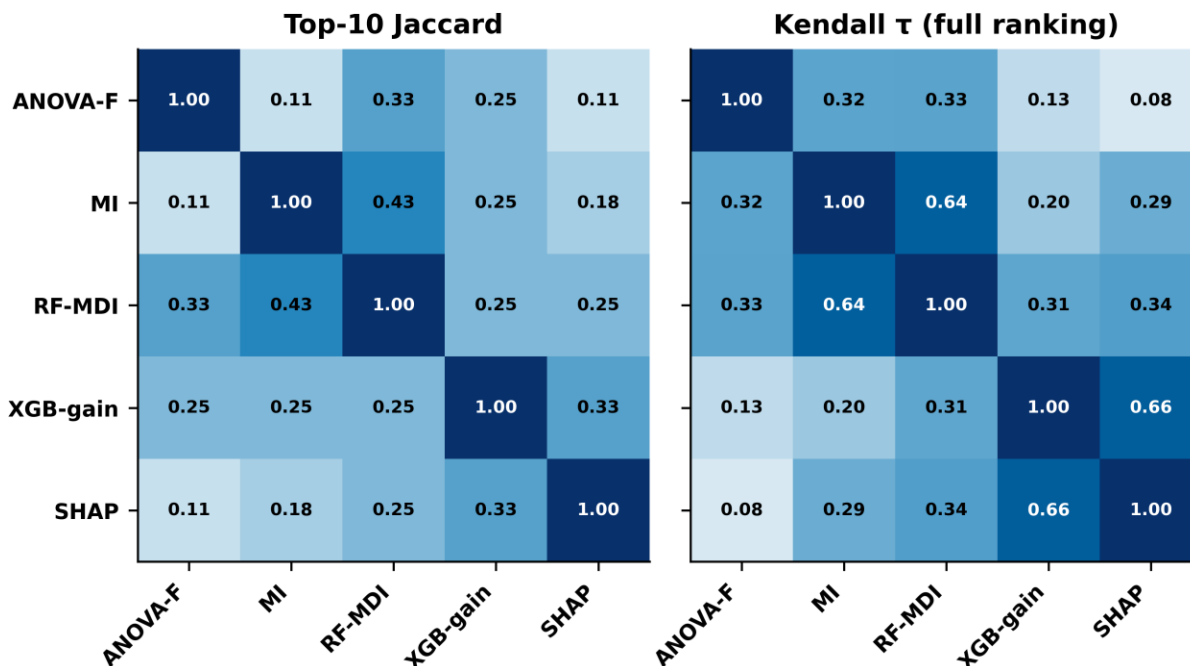


Figure 5. Agreement between the five feature-ranking methods on CIC-IDS2017 training data: Jaccard index of the top-k sets (left) and Kendall's τ between full rankings (right).

Table 4. Feature-selection outcome per dataset and task (reference model LightGBM, validation split).

Dataset	Task	Ranker	k	All feat.	Val F1 (all)	Val F1 (sel.)
CIC-IDS2017	multi	shap	10	69	0.9933	0.9937
CIC-IDS2017	binary	xgb_importance	10	69	0.9980	0.9971
CICIoT2023	multi	xgb_importance	20	39	0.7152	0.6994
CICIoT2023	binary	xgb_importance	20	39	0.8972	0.8990
UNSW-NB15	multi	xgb_importance	10	58	0.5722	0.6119
UNSW-NB15	binary	xgb_importance	20	58	0.9233	0.9247

Whether a reduced subset matches the full feature set depends on the dataset. On the CIC-IDS2017 test split, LightGBM scored 0.9928 macro-F1 with all 69 transformed features and 0.9942 with ten. On UNSW-NB15, the ten selected features improved macro-F1 for 5 of the seven models; Light GBM rose from 0.5741 to 0.6224. On CICIoT2023 the 20-feature subset cost Light GBM 0.7108 $\rightarrow$ 0.7012. Logistic regression and the MLP lost the most on CIC-IDS2017 and UNSW-NB15; logistic regression on CIC-IDS2017 fell from 0.8112 to 0.4564. Selection reduced LightGBM training time to 43% (CIC-IDS2017) and 50% (UNSW-NB15) of the full-feature time, and SHAP explanation time to 52% and 32%, respectively (Supplementary Table S4).

Model performance

Table 5 and Figure 6 report multiclass test performance. On CIC-IDS2017, LightGBM reached the highest macro-F1 (0.9942, 95% CI [0.9920, 0.9963]), followed by XGBoost (0.9924) and random forest (0.9870). The overlap of these intervals already suggests that the ranking among the three ensembles is not reliable (Section 11). Logistic regression (0.4564) and the MLP (0.8277) trailed clearly. Accuracy hides these differences: every model except logistic regression exceeded 0.96 accuracy (supplementary table S2).

Scores were much lower on the other datasets. The best macro-F1 on UNSW-NB15 was 0.6224 (LightGBM) and on CICIoT2023 0.7012 (LightGBM). The per-class results explain why. On CICIoT2023, the best model identified Mirai traffic almost perfectly (F1 0.998) but confused DoS with DDoS (DoS F1 0.603) and struggled with the small Web (F1 0.439) and BruteForce (0.415) families. This matches the high label-conflict count of Section 8.1. On UNSW-NB15, the Analysis (F1 0.13), Backdoor (0.22) and DoS (0.33) families were rarely recognised, while benign and Generic traffic exceeded 0.9. In the binary task, scores were high on CIC-IDS2017 (best F1 0.9974) and UNSW-NB15 (0.9193) (supplementary table S1).

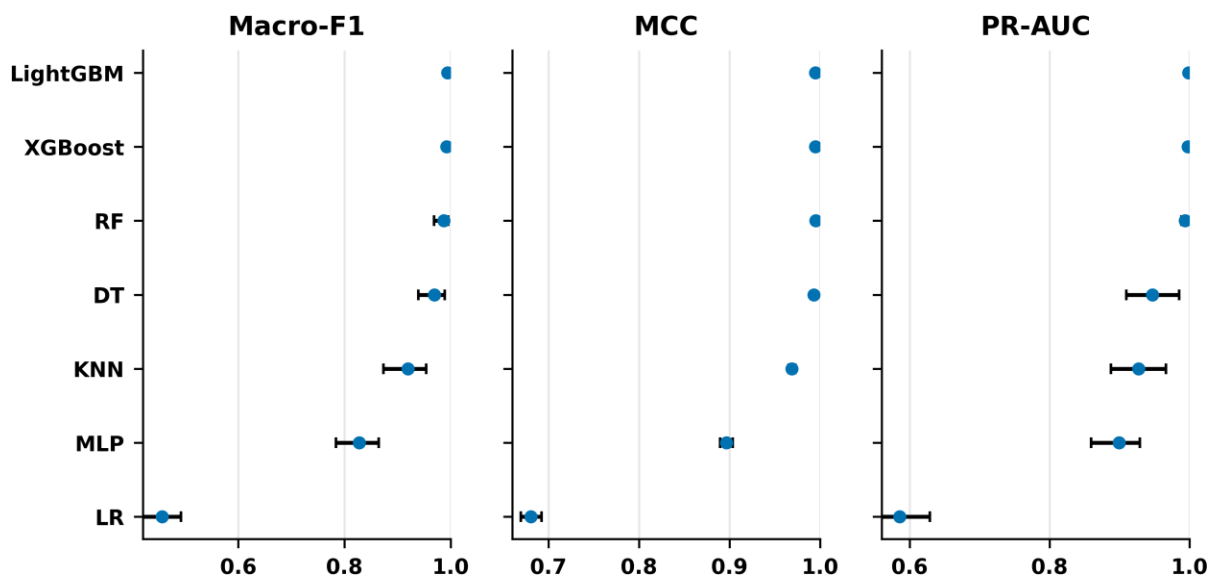


Figure 6. Test-set performance of 7 models on CIC-IDS2017 (multiclass task); bars are 95% bootstrap CIs. Highest macro-F1: LightGBM (0.9942); whether differences are significant is assessed separately.

Table 5. Multiclass test-set performance (selected features). FPR/FNR are attack-vs-benign rates derived from the multiclass predictions.

Dataset	Model	Macro-F1	F1 95% CI	MCC	PR-AUC	FPR	FNR	Train s	µs/flow
CIC-IDS2017	LR	0.4564	[0.4093, 0.4922]	0.6808	0.5858	0.0214	0.3938	17.9519	0.4254
CIC-IDS2017	DT	0.9696	[0.9391, 0.9891]	0.9929	0.9468	0.0018	0.0044	0.3369	0.0712
CIC-IDS2017	RF	0.9870	[0.9689, 0.9953]	0.9952	0.9935	0.0011	0.0037	5.9354	7.0467
CIC-IDS2017	XGBoost	0.9924	[0.9895, 0.9951]	0.9949	0.9978	0.0014	0.0031	7.4434	17.0266
CIC-IDS2017	LightGBM	0.9942	[0.9920, 0.9963]	0.9947	0.9985	0.0013	0.0037	8.6581	115.1923
CIC-IDS2017	KNN	0.9198	[0.8733, 0.9542]	0.9689	0.9273	0.0078	0.0190	0.1286	155.1334
CIC-IDS2017	MLP	0.8277	[0.7838, 0.8648]	0.8965	0.8991	0.0233	0.0702	69.6184	2.0434
CICIoT2023	LR	0.5793	[0.5643, 0.5942]	0.5716	0.6332	0.4526	0.0070	151.6819	0.2978
CICIoT2023	DT	0.6395	[0.6251, 0.6531]	0.6138	0.5792	0.2912	0.0081	12.6945	0.1687
CICIoT2023	RF	0.6875	[0.6713, 0.7037]	0.6371	0.7315	0.1906	0.0088	139.9648	18.2208
CICIoT2023	XGBoost	0.6962	[0.6826, 0.7103]	0.6434	0.7585	0.2175	0.0066	52.7779	25.5086
CICIoT2023	LightGBM	0.7012	[0.6865, 0.7158]	0.6454	0.7607	0.1988	0.0071	53.3131	111.0708
CICIoT2023	KNN	0.5819	[0.5689, 0.5965]	0.5866	0.5109	0.3895	0.0095	0.8253	935.1882
CICIoT2023	MLP	0.6298	[0.6152, 0.6443]	0.6186	0.6902	0.3181	0.0084	518.6318	2.6327
UNSW-NB15	LR	0.3534	[0.3406, 0.3688]	0.5592	0.4149	0.1253	0.2265	13.6532	0.2654
UNSW-NB15	DT	0.5829	[0.5652, 0.5996]	0.7017	0.6034	0.0644	0.1184	0.2387	0.1469

UNSW-NB15	RF	0.6214	[0.6070, 0.6344]	0.724 4	0.678 5	0.066 8	0.103 5	5.3386	15.5670
UNSW-NB15	XGBoost	0.6096	[0.5942, 0.6251]	0.731 8	0.683 7	0.058 1	0.105 0	11.5329	26.7746
UNSW-NB15	LightGBM	0.6224	[0.6077, 0.6354]	0.732 8	0.692 4	0.061 9	0.103 0	22.0589	274.4523
UNSW-NB15	KNN	0.6084	[0.5916, 0.6223]	0.709 0	0.604 9	0.071 0	0.110 4	0.1318	25.6295
UNSW-NB15	MLP	0.4629	[0.4430, 0.4834]	0.673 4	0.533 4	0.068 6	0.133 4	44.8319	2.0971

From an operational point of view, the attack-vs-benign false-positive rate of the best CIC-IDS2017 model was 0.0013, but it was 0.199 on CICIoT2023, where benign recall is only 80.1%. KNN trained fastest but was expensive at prediction time (up to 935 μ s per flow on CICIoT2023). Multiclass LightGBM was also comparatively slow to predict (274 μ s per flow on UNSW-NB15), because it evaluates one tree per class in every boosting round.

Explainability Analysis

Figure 7 shows the global TreeSHAP explanation of the CIC-IDS2017 LightGBM model. The three most influential features were Destination Port, Init_Win_bytes_forward, Flow Duration. The dominance of Destination Port is a warning sign rather than a success: in CIC-IDS2017, many attacks target fixed service ports, so the model may be learning testbed-specific shortcuts that will not generalise; such spurious correlations are a recognised pitfall of security ML [6]. On

UNSW-NB15 the top features were smean, sbytes, dbytes, and on CICIoT2023 Number, Tot sum, AVG. Figure 8 decomposes one correctly detected attack into additive feature contributions, the view an analyst would use to check a single alert.

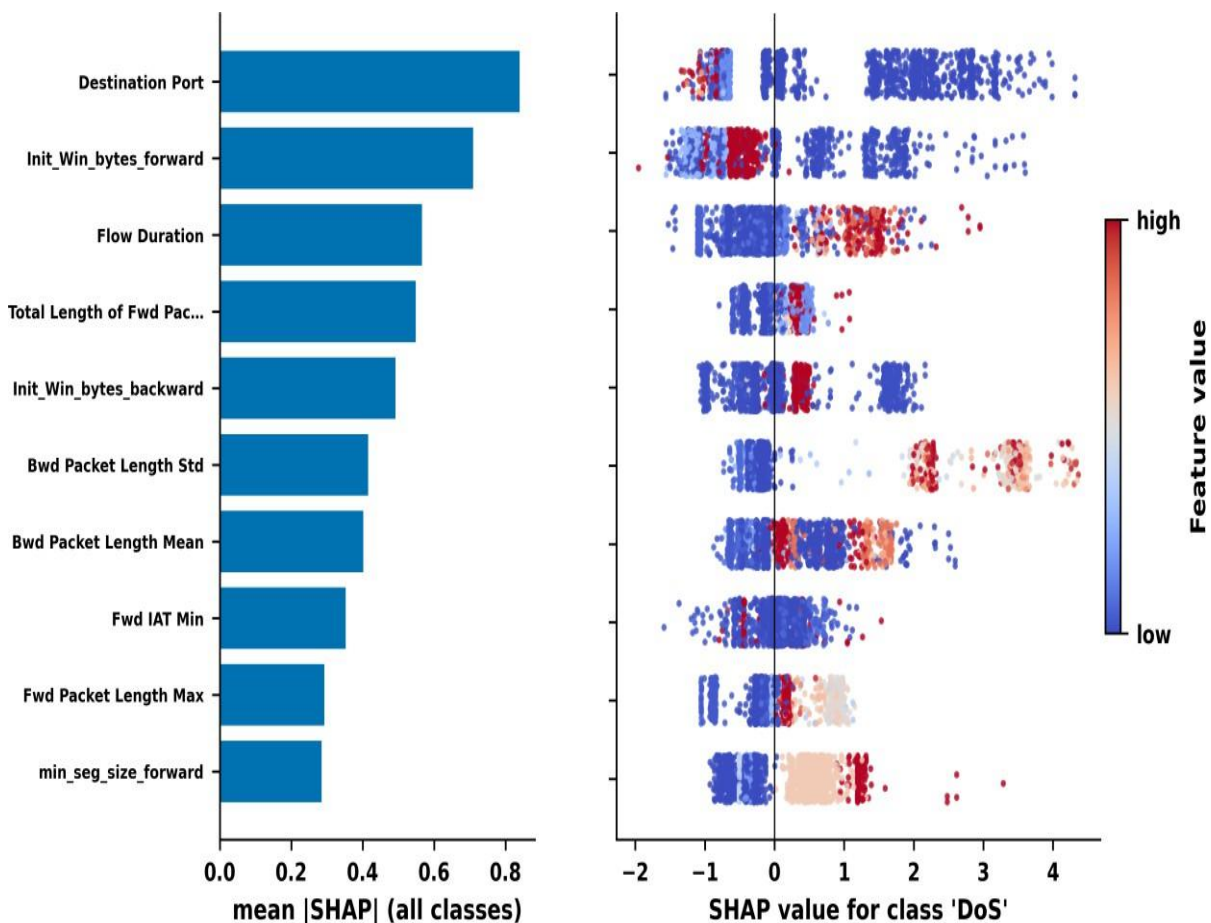


Figure 7. Global TreeSHAP explanation of LightGBM on 1,610 CIC-IDS2017 test flows: mean absolute SHAP value (left) and per-flow SHAP values for class 'DoS' coloured by feature value (right). SHAP values describe the model, not causal attack mechanisms.

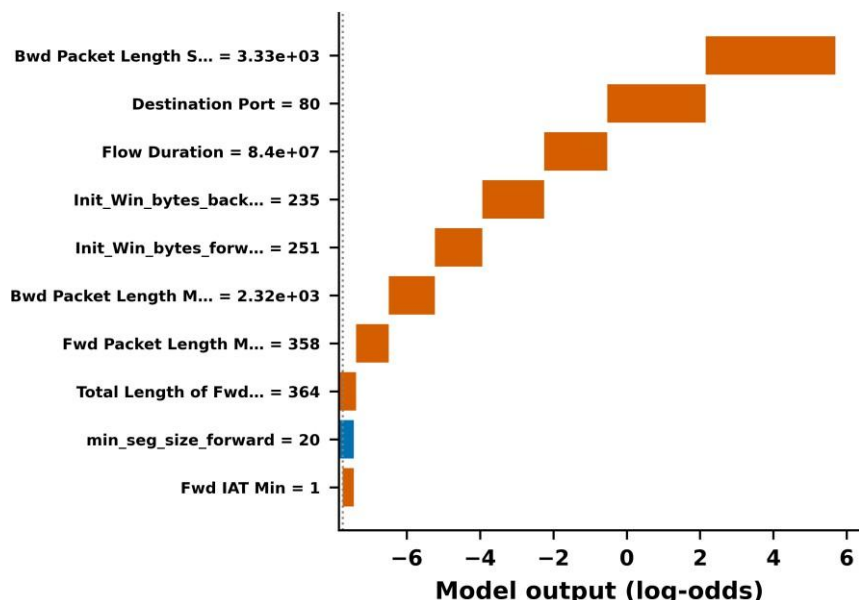


Figure 8. Local TreeSHAP explanation of one CIC-IDS2017 test flow predicted as 'DoS' by LightGBM. Bars show each feature's additive contribution from the base value -7.75 (dotted) to the output 5.69; red pushes towards 'DoS'.

Table 6 compares SHAP and LIME on 40 stratified test flows per run. SHAP was 22 to 156 times cheaper per explanation than LIME: 3.9 ms against 324 ms on CIC-IDS2017, and 12.5 ms against 406 ms on CICIoT2023. LIME's top-5 sets varied between random seeds (mean pairwise Jaccard 0.44-0.74 across runs). SHAP's global top-5 was identical across retrained models in 4 of 6 runs; the exceptions were UNSW-NB15 multiclass (0.67) and CICIoT2023 binary (0.59). On the same instance the two methods often disagreed: top-5 agreement ranged from 0.27 to 0.52.

Deletion fidelity separates the methods more clearly. Replacing SHAP's top-5 features by training medians reduced the predicted-class probability significantly more than replacing five random features in 6 of 6 runs (paired Wilcoxon; e.g., mean drop 0.598 vs 0.262 on CICIoT2023, $p < 10^{-4}$). LIME's top features beat the random baseline in 3 of 6 runs; they did not on CIC-IDS2017 binary ($p = 0.145$) or on either CICIoT2023 task ($p = 0.139$ and 0.979). For tree ensembles, then, TreeSHAP was cheaper and, under this deletion test, more faithful to the model. The two stability measures are not directly comparable (SHAP across retrained models, LIME across random seeds), and in UNSW-NB15 multiclass SHAP's retraining stability was slightly below LIME's seed stability. LIME remains model-agnostic, which SHAP's tree-specific algorithm is not. Neither method's output should be read as a causal account of an attack.

Table 6. SHAP versus LIME (top-5 features).

Dataset	Task	Model	SHAP ms	LIME ms	LIME stab.	SHA P stab.	Agre e.	Del. SHA P	Del. LIME	Del. rand.	p SHA P	p LIME
CIC-IDS2017	multi	LightGBM	3.9461	324.4708	0.7093	1.0000	0.5193	0.8735	0.7379	0.6616	1.00e-10	1.93e-04
CIC-IDS2017	binary	LightGBM	1.1431	113.0389	0.7445	1.0000	0.4688	0.5204	0.1882	0.2349	3.95e-08	0.1447
CICIoT2023	multi	LightGBM	12.4638	405.9248	0.6713	1.0000	0.2727	0.5983	0.3649	0.2622	6.36e-06	0.1387
CICIoT2023	binary	LightGBM	4.0733	209.8655	0.4380	0.5873	0.2775	0.2632	0.1846	0.2155	0.0424	0.9786
UNSW-NB15	multi	XGBoost	7.4497	162.2904	0.6917	0.6667	0.4518	0.5207	0.4768	0.3837	0.0012	0.0208
UNSW-NB15	binary	LightGBM	1.3739	213.6406	0.6461	1.0000	0.5178	0.3981	0.3437	0.0724	4.15e-06	0.0061

Intelligent Visualization Analysis

ICVRE fired between 16 and 18 of its 20 rules per run and planned 12 figures each time; all planned figures were rendered without manual edits (Table 7). The plans differed between datasets in ways that follow the profiles. Both CIC-IDS2017 plans included a capture-day timeline (only CIC-IDS2017 has a temporal attribute), and the multiclass plan a class chart with a logarithmic axis (imbalance ratio above 100). The UNSW-NB15 plans omitted the missing-value chart, because UNSW-NB15 has no missing or infinite values. In the multiclass runs (imbalance ratios above 100) the PR curve was ranked above the ROC curve; in the three binary runs (imbalance ratios between about 1 and 27) the ROC curve ranked

higher. In CICIOT2023 binary, where attacks outnumber benign flows about 27 to 1, both curves were planned but the ROC curve still ranked first; a PR-first order would arguably suit that case better, so the imbalance weighting of these rules is a design choice rather than a validated setting. Every omission is recorded with its reason (supplementary table S6).

Against the literature-derived reference, 47 of the 59 scored figure recommendations (79.7%) used a chart family that published XAI-IDS papers used for the same task. Per run, agreement ranged from 70% to 90%. ICVRE covered 67%-75% of the twelve analytic tasks found in the corpus. The tasks left uncovered were computational cost (reported by the framework as tables, never as figures), feature relationships and feature effects (scatter and dependence plots usually fell below the cut-off), and temporal patterns outside CIC-IDS2017. Disagreements came from three recommendations: the ranking-agreement heatmap, which has no counterpart in the corpus; PR curves, where the corpus used only ROC curves and TPR/FPR bars; and the CIC-IDS2017 timeline, where the corpus used a heatmap. These figures show consistency with published practice, not recommendation accuracy; the expert-rating protocol remains pending execution.

Table 7. ICVRE evaluation against the literature-derived reference (44 coded figures from 7 papers).

Dataset	Task	Rules fired	Planned	Rendered	Suppressed	Score d	Agreein g	Agreemen t	Coverag e
CIC-IDS2017	multi	18	12	12	8	10	7	0.7000	0.7500
CIC-IDS2017	binary	18	12	12	8	9	7	0.7778	0.6667
CICIOT2023	multi	17	12	12	8	10	8	0.8000	0.6667
CICIOT2023	binary	17	12	12	8	10	8	0.8000	0.6667
UNSW-NB15	multi	16	12	12	8	10	8	0.8000	0.7500
UNSW-NB15	binary	16	12	12	8	10	9	0.9000	0.7500

One of the figures ICVRE planned for every run was a PCA/UMAP projection (rule R07, triggered by more than ten dimensions). Figure 9 shows it for CIC-IDS2017 test flows. In the PCA view most classes overlap near the origin, and DoS and DDoS flows spread along one principal direction. UMAP splits benign traffic into several separate clusters and places DoS, DDoS and port-scan flows in compact groups of their own, while botnet and web-attack flows lie next to or inside benign clusters. This matches the lower per-class scores of those small families. The projections are descriptive and are not used by any model.

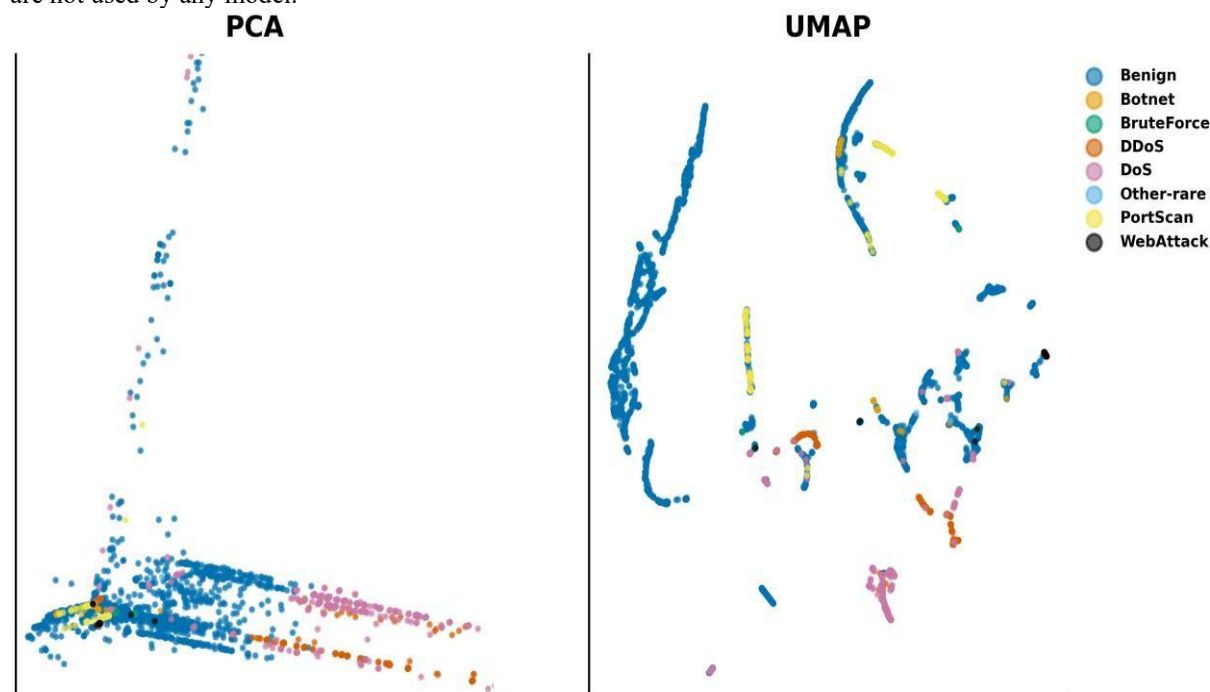


Figure 9. Two-dimensional projections of 4,000 randomly sampled CIC-IDS2017 test flows (standardised features): PCA (left; first two components explain 47.6% of variance) and UMAP (right); axes span the 0.5-99.5th percentiles.

Statistical Validation

Table 8 summarises the tests on 10-fold CV macro-F1. In every run the Friedman test rejected the hypothesis that all seven models perform equally (largest $p = 5.4e-08$), and Kendall's W was between 0.74 and 0.93, indicating a consistent model ordering across folds. After Holm correction, between 11/21 and 18/21 pairwise differences were significant. The pattern was the same in all six runs (three datasets $\times$ two tasks): the three ensembles (RF, XGBoost, LightGBM) never differed significantly from one another, but each differed significantly from logistic regression and from the MLP. Their differences from KNN and the single decision tree were significant in most runs but not all; on UNSW-NB15 multiclass, for example, neither DT nor KNN was separable from the ensembles.

The comparisons that a typical benchmark paper would headline were not significant. On CIC-IDS2017 multiclass, the two best models by CV were LightGBM and RF. Their mean CV difference was 0.0264 [0.0013, 0.0622] (bootstrap 95% CI), the effect size was large ($r_{rb} = -0.71$, negative because RF is the first model of the pair), and yet the Holm-adjusted Wilcoxon test was not significant ($p = 0.195$); McNemar's test on the held-out test set also found no difference ($p = 0.54$). With ten folds the smallest possible two-sided Wilcoxon p -value is about 0.002, so after Holm correction over 21 pairs only differences that hold in nearly every fold remain significant. For XGBoost versus LightGBM, the CV difference was within ± 0.02 and its interval contained zero on UNSW-NB15 (0.0033 [-0.0092, 0.0186]), CICIoT2023 (0.0078 [-0.0018, 0.0191]) and CIC-IDS2017 binary (0.0001 [-0.0005, 0.0007]). McNemar's test was nevertheless significant for the top-two pair on CICIoT2023 multiclass ($p = 2.1e - 07$) and on UNSW-NB15 binary ($p = 9.4e - 05$). With about 24,000 test flows, McNemar detects small differences in error patterns that do not translate into a reliable macro-F1 gain. We therefore do not claim that any single model is best. The evidence supports only the weaker statement that gradient-boosted and bagged tree ensembles outperform the linear and neural baselines in every run, and the single decision tree and KNN in most runs, on these data.

Table 8. Statistical validation on 10-fold CV macro-F1 (Friedman omnibus, pairwise Wilcoxon with Holm correction, rank-biserial effect size) and McNemar on the test split.

Dataset	Task	χ^2	p (Fr.)	W	Sig. pairs	Top-2 (CV)	$\Delta F1$ [95% CI]	p_adj (W)	r_rb	p (McN.)
CIC-IDS2017	multi	54.6429	5.47e-10	0.9107	16/21	LightGBM vs RF	0.0264 [0.0013, 0.0622]	0.1953	-0.7091	0.5413
CIC-IDS2017	binary	56.5606	2.24e-10	0.9309	18/21	LightGBM vs XGBoost	0.0001 [-0.0005, 0.0007]	1.0000	0.0000	1.0000
CICIoT2023	multi	51.9857	1.88e-09	0.8664	14/21	LightGBM vs XGBoost	0.0078 [-0.0018, 0.0191]	0.8262	-0.4182	2.14e-07
CICIoT2023	binary	50.2286	4.23e-09	0.8371	13/21	LightGBM vs XGBoost	0.0053 [-0.0016, 0.0124]	0.9668	-0.3818	0.6338
UNSW-NB15	multi	44.7000	5.37e-08	0.7450	11/21	XGBoost vs LightGBM	0.0033 [-0.0092, 0.0186]	1.0000	-0.1273	0.8561
UNSW-NB15	binary	53.4429	9.55e-10	0.8907	15/21	LightGBM vs XGBoost	0.0010 [-0.0005, 0.0024]	0.5801	-0.4909	9.45e-05

The significance heatmap produced by ICVRE for CIC-IDS2017 (supplementary figure S1) shows the same pattern: dense blocks of significant differences between families and non-significant cells among the three ensembles.

Ablation Study

Table 9 and Figure 10 isolate the contribution of each component in the multiclass task. Feature selection (A $\rightarrow$ B) cut the number of features from 69 to 10 on CIC-IDS2017, from 58 to 10 on UNSW-NB15 and from 39 to 20 on CICIoT2023. The best macro-F1 moved from 0.9941 to 0.9942, from 0.5741 to 0.6224 and from 0.7108 to 0.7012, respectively. Selection therefore helped on UNSW-NB15, was neutral on CIC-IDS2017 and cost about one point on CICIoT2023. Its cost in pipeline time depends on whether ranking time exceeds the training time saved: B was faster than A on CICIoT2023 (1169 s vs 1394 s) but slower on CIC-IDS2017 (341 s vs 230 s).

The XAI engine (B $\rightarrow$ C) added 148 s on CIC-IDS2017, 137 s on UNSW-NB15 and 366 s on CICIoT2023. This covers TreeSHAP on up to 2,000 flows, LIME on 40 flows with five seeds each, and three model retrains for the stability test. The complete framework (C $\rightarrow$ D) added statistical validation, ICVRE-planned figures and tables, and a hashed provenance record; the pipeline time rose to 815 s, 900 s and 4700 s. The 10-fold CV behind the statistical tests is the

dominant cost, especially on CICIoT2023, where SMOTE enlarges every training fold. By construction, C and D do not change predictive scores. Their value lies in outputs that A and B lack: explanations with measured stability and fidelity, significance tests, 12 figures and 8 tables per run, and a record of every artifact's hash.

Table 9. Component ablation. Pipeline time excludes hyperparameter search (identical for all configurations); C and D reuse B's models, so predictive scores are unchanged by design.

Dataset	Config	Features	Best model	Macro-F1	MCC	Pipeline s	XAI s	Figures	Tables
CIC-IDS2017	A: ML, all features	69	XGBoost	0.9941	0.9953	229.5620	0.0000	0	0
CIC-IDS2017	B: + feature selection	10	LightGBM	0.9942	0.9947	341.1813	0.0000	0	0
CIC-IDS2017	C: + XAI	10	LightGBM	0.9942	0.9947	489.3040	148.1226	0	0
CIC-IDS2017	D: complete framework	10	LightGBM	0.9942	0.9947	814.6828	148.1226	12	8
CICIoT2023	A: ML, all features	39	LightGBM	0.7108	0.6453	1394.1204	0.0000	0	0
CICIoT2023	B: + feature selection	20	LightGBM	0.7012	0.6454	1168.5186	0.0000	0	0
CICIoT2023	C: + XAI	20	LightGBM	0.7012	0.6454	1534.4159	365.8973	0	0
CICIoT2023	D: complete framework	20	LightGBM	0.7012	0.6454	4700.1272	365.8973	12	8
UNSW-NB15	A: ML, all features	58	LightGBM	0.5741	0.7242	225.8046	0.0000	0	0
UNSW-NB15	B: + feature selection	10	LightGBM	0.6224	0.7328	376.8878	0.0000	0	0
UNSW-NB15	C: + XAI	10	LightGBM	0.6224	0.7328	514.1613	137.2734	0	0
UNSW-NB15	D: complete framework	10	LightGBM	0.6224	0.7328	900.4340	137.2734	12	8

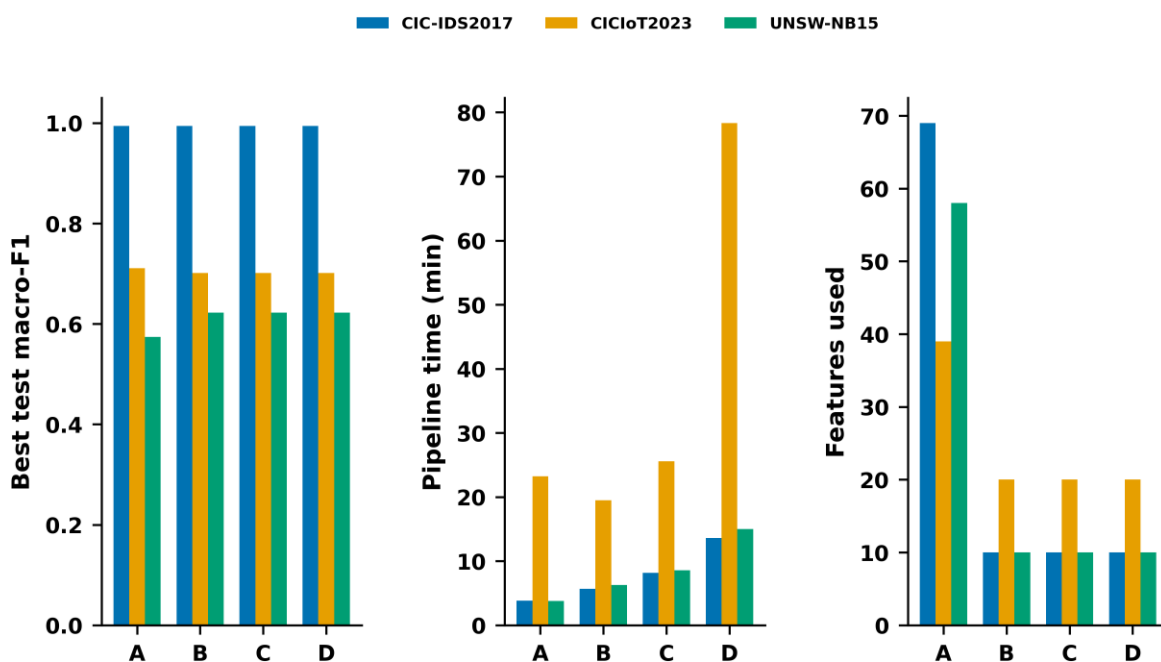


Figure 10. Component ablation (multiclass): A = models on all features, B = + feature selection, C = + XAI, D = complete framework (statistics, ICVRE figures, tables, provenance). C and D add outputs, not accuracy (see the ablation table).

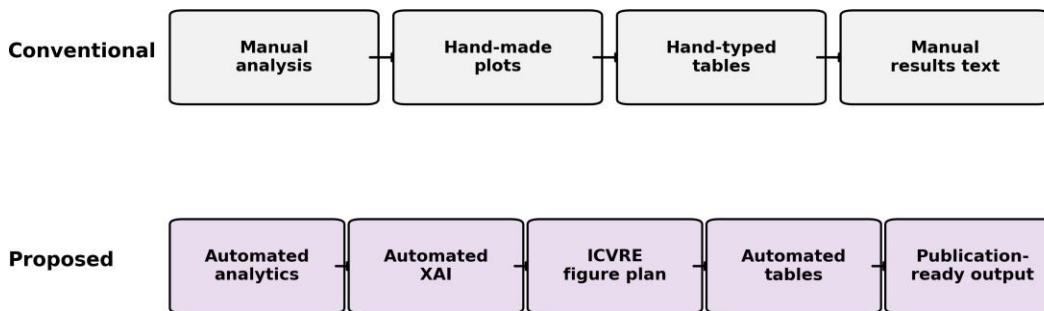
Publication workflow and reproducibility

Table 10 and Figure 11 summarise the automated workflow. Each of the 6 dataset-task runs was started with one command and required no manual edits. Together they produced 72 figures (each in four formats, with manifest entries and captions generated from the data) and 48 tables (each in five formats), and recorded 360 hashed artifacts. Wall-clock time per run ranged from 11 to 112 minutes on the two-vCPU machine. Figure rendering took 74 s per run on average, and table generation under one second. How long an analyst would need to produce the same outputs by hand was not measured (Experiment pending execution), so we make no claim about time savings.

The re-execution of UNSW-NB15 (binary) with the identical configuration reproduced 49 of 61 compared artifacts byte for byte. All differences were traced. Four are the files of the XAI-comparison figure, which plots measured explanation times. The remaining eight come from random forest's predicted probabilities: parallel tree evaluation on two threads sums floating-point values in a varying order, which changed its ROC-AUC and PR-AUC by at most $2.7e-08$. This propagated into the ROC figure, two vector files of the model-comparison figure and the two metric tables. Class predictions and every label-based metric were identical (maximum difference 0.0), as were the CV scores, all statistical tests and the SHAP-LIME comparison table. Independently, CIC-IDS2017 multiclass was executed twice with the same configuration (the second time to regenerate a corrected timeline figure) and gave identical test scores for all seven models.

Table 10. Publication-workflow measurements for the automated workflow. The conventional manual workflow was not timed (Experiment pending execution: timed user study).

Dataset	Task	Wall time s	Figure s	Table s	Figure gen. s	Table gen. s	Artifacts hashed	Manual edits
CIC-IDS2017	multi	1634.1665	12	8	72.7660	0.3296	60	0
CIC-IDS2017	binary	688.7222	12	8	75.4300	0.2986	60	0
CICIoT2023	multi	6716.4015	12	8	73.4600	0.2917	60	0
CICIoT2023	binary	699.8932	12	8	73.1620	0.3068	60	0
UNSW-NB15	multi	1907.1300	12	8	73.9820	0.2828	60	0
UNSW-NB15	binary	811.4390	12	8	73.1560	0.6887	60	0



runs: 6 · commands: 6 · figures: 72 · tables: 48 · median wall time (min): 20.4 · manual edits: 0

Figure 11. Conventional versus proposed publication workflow. The bottom line lists quantities measured in this study for the proposed workflow (see the workflow table); the conventional workflow was not timed.

Cross-Dataset Validation

Experiments A, C and D (independent evaluation). Applied unchanged to each dataset, the framework produced the pattern of Figure 12 (left): high macro-F1 on CIC-IDS2017 (Experiment A), and markedly lower values on CICIoT2023 (C) and UNSW-NB15 (D), with the tree ensembles at the top on all three. The binary task narrows the gap: the best F1 was 0.9974 on CIC-IDS2017, 0.9029 on CICIoT2023 and 0.9193 on UNSW-NB15 (Table 11). The benign false-positive rate on CICIoT2023 remained high (0.199), because benign traffic makes up only 3.6% of that subset.

Experiment B (transfer on aligned features). Transfer failed in every direction (Figure 12, right; Table 11). Trained on CIC-IDS2017 and tested on UNSW-NB15, the three ensembles labelled almost every flow as benign: at most 0.012 attack F1, with MCC between -0.016 and 0.031. Trained on UNSW-NB15 and tested on CIC-IDS2017, they flagged between half and two-thirds of all flows as attacks, most of them benign, and MCC was negative. The CIC-IDS2017 → CICIoT2023 models predicted no attacks at all (F1 = 0). The reverse models flagged nearly every CIC-IDS2017 flow as an attack (MCC ≈ 0). The same aligned features supported good within-target detectors: within-target F1 was 0.894 on

UNSW-NB15 and 0.955 on CIC-IDS2017 with the same seven features. The collapse therefore reflects distribution shift between testbeds and extractors, not a lack of information in the aligned features. This agrees with earlier cross-dataset studies on CIC data [8, 9, 48]. PR-AUC is misleading in this setting: on CICIoT2023, where 96% of test flows are attacks, the no-skill PR-AUC already equals that prevalence, so the CIC-IDS2017 models reached PR-AUC values above 0.9 although none of them flagged a single attack at the 0.5 threshold. MCC is therefore the primary transfer metric.

Table 11. Cross-dataset validation (binary task). Experiment B uses only semantically aligned features; each transfer row lists the within-target score on the same features.

Exp .	Train	Test	Features	Model	F1 (attack)	MCC	PR-AUC	Note
A	CIC-IDS2017	CIC-IDS2017	10 (selected)	XGBoost	0.9959	0.9949	0.9996	within-dataset, full framework
C	CICIoT2023	CICIoT2023	20 (selected)	LightGBM	0.9933	0.8059	0.9998	within-dataset, full framework
D	UNSW-NB15	UNSW-NB15	20 (selected)	LightGBM	0.9150	0.8386	0.9812	within-dataset, full framework
B	CIC-IDS2017	UNSW-NB15	7 aligned (moderate)	RF	0.0000	0.0000	0.4817	within-target on same features: F1 0.8829, MCC 0.7761
B	CIC-IDS2017	UNSW-NB15	7 aligned (moderate)	XGBoost	0.0115	0.0311	0.5413	within-target on same features: F1 0.8920, MCC 0.7927
B	CIC-IDS2017	UNSW-NB15	7 aligned (moderate)	LightGBM	0.0012	-0.0162	0.5746	within-target on same features: F1 0.8943, MCC 0.7970
B	UNSW-NB15	CIC-IDS2017	7 aligned (moderate)	RF	0.1837	-0.2679	0.1271	within-target on same features: F1 0.9527, MCC 0.9418
B	UNSW-NB15	CIC-IDS2017	7 aligned (moderate)	XGBoost	0.2606	-0.0387	0.1591	within-target on same features: F1 0.9541, MCC 0.9437
B	UNSW-NB15	CIC-IDS2017	7 aligned (moderate)	LightGBM	0.2491	-0.0439	0.2136	within-target on same features: F1 0.9551, MCC 0.9448
B	CIC-IDS2017	CICIoT2023	5 aligned (weak)	RF	0.0000	0.0000	0.9613	within-target on same features: F1 0.9895, MCC 0.6854

B	CIC-IDS2017	CICIoT2023	5 aligned (weak)	XGBoost	0.0000	0.0000	0.9326	within-target on same features: F1 0.9858, MCC 0.7070
B	CIC-IDS2017	CICIoT2023	5 aligned (weak)	LightGBM	0.0000	0.0000	0.9757	within-target on same features: F1 0.9832, MCC 0.6987
B	CICIoT2023	CIC-IDS2017	5 aligned (weak)	RF	0.3147	0.0031	0.1448	within-target on same features: F1 0.9077, MCC 0.8863
B	CICIoT2023	CIC-IDS2017	5 aligned (weak)	XGBoost	0.3143	-0.0039	0.1497	within-target on same features: F1 0.9086, MCC 0.8885
B	CICIoT2023	CIC-IDS2017	5 aligned (weak)	LightGBM	0.3165	0.0385	0.1553	within-target on same features: F1 0.9073, MCC 0.8869

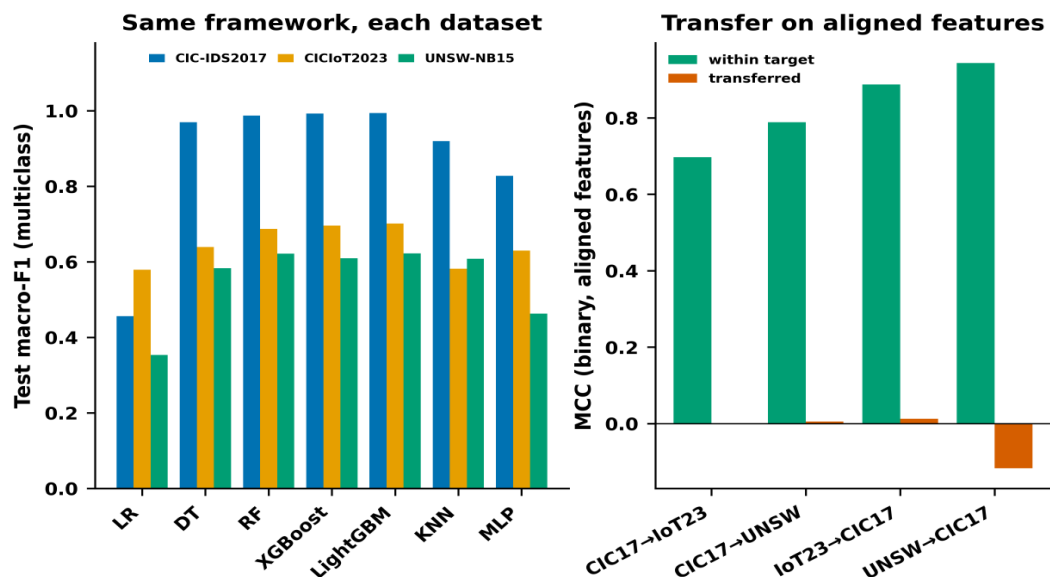


Figure 12. Cross-dataset comparison. Left: test macro-F1 of each model when the full framework is applied independently to each dataset (Experiments A, C, D). Right: MCC of models trained on one dataset and tested on another using only semantically aligned features, next to the within-target MCC on the same features (Experiment B; mean over RF, XGBoost and LightGBM).

DISCUSSION

What the framework adds. Relative to the studies in Table 1, the framework's distinctive outputs are not higher scores but evidence that is usually missing. Every run of it produced significance tests with effect sizes, a quantitative SHAP-LIME comparison, an evidence-gated figure plan with recorded omissions, and hashed artifacts. These outputs changed the conclusions we can draw. Mean test scores alone would suggest that LightGBM was the best model, as it had the highest macro-F1 in 5 of 6 runs; the tests show that it is statistically indistinguishable from XGBoost and random forest

everywhere. This agrees with long-standing advice against ranking classifiers by mean scores [14, 15]. It also puts into perspective many single-split comparisons in the XAI-IDS literature that report a new best model [40, 51, 54].

Data quality dominates model choice. The largest effects in this study came from the data, not the models. De-duplication removed between about a tenth and more than a third of each dataset, and label conflicts, identical feature vectors with different labels, affected 132,347 CICIOT2023 rows. These conflicts explain why DoS and DDoS are confused and why CICIOT2023 macro-F1 stays near 0.7, far below the accuracy figures often reported for this dataset [41, 43]. Accuracy and macro-F1 measure different things, so these numbers are not directly comparable; our point is that duplicates and conflicts must be reported for either number to be interpretable. The finding extends earlier audits of CIC-IDS2017 [2, 11, 12] to CICIOT2023 and supports the leakage taxonomy of Kapoor and Narayanan [7].

Explanations need their own validation. SHAP and LIME often disagreed on the same flow, as Hermosilla et al. [23] and Nugraha et al. [24] also observed. Our deletion test gives a way to decide between them for a given model: SHAP's top features mattered more to the model than random features in 6 of 6 runs, LIME's in 3. The prominence of Destination Port in CIC-IDS2017 explanations illustrates why explanations should be read as a diagnostic of the model, not of the attack. A model that relies on the port number may be exploiting how the testbed scheduled attacks, an instance of the spurious correlations that Arp et al. [6] list among common pitfalls in security ML. The framework surfaces such cases; it cannot decide whether a feature is legitimate.

Figure planning. ICVRE's plans were consistent with published practice for 80% of the scored figure recommendations. Its main departures (PR curves under imbalance, a ranking-agreement view, a timeline instead of a heatmap) are deliberate and follow reporting advice for imbalanced data. ICVRE differs from general recommenders [27, 28, 30] in taking model outputs, explanations and tests as inputs, and from security visual-analytics systems [60, 61] in targeting reporting rather than exploration. Its rules are transparent and cheap, but they are hand-designed. Learned recommenders or LLM-based tools [31, 32] could propose views the rules miss, at the price of explanations that are harder to audit.

Automation and integrity. The workflow experiment shows what automation delivered in this study: one command per run, 72 figures and 48 tables across 6 runs, no manual edits and 360 hashed artifacts. It does not show how much time this saves compared with a manual workflow, because we did not time analysts. Unlike LLM paper generators [33], the framework does not write claims. It produces the evidence and the provenance that a human author, or a reviewer, can check, which addresses the integrity concerns raised about AI-generated scientific content [74, 75].

LIMITATIONS

Sampling and subsets. Models were trained on 120,000-flow stratified samples, and CICIOT2023 on three of 63 merged files, because of the 2-vCPU, 7 GB test machine. Minority retention slightly over-represents small classes, which affects precision estimates. Results may differ with full data.

Single seed and optimistic tests. All runs used seed 42. The statistical tests use folds of one dataset that share training data, so their p-values are optimistic; they do not replace replication on independent data.

Limited tuning. Random search with six candidates on a 20,000-flow subsample may leave some models, notably the MLP and KNN, under-tuned. The RBF-kernel SVM was not evaluated.

Weak cross-dataset alignment. Transfer was evaluated on 5-7 manually aligned features, and CICIOT2023 alignment is approximate. Transfer results should be read as evidence about distribution shift, not as deployable cross-network detectors.

Explanation evaluation. Deletion fidelity with median replacement can create out-of-distribution inputs; it measures reliance of the model on features, not correctness of explanations. LIME was evaluated with one kernel width and 2,000 perturbation samples.

ICVRE evaluation. The reference set was coded from captions of seven papers by one annotator, and published choices are not a gold standard. The expert-rating protocol and the timed comparison with a manual workflow are Experiment pending execution. The dashboard was tested for correctness but not evaluated with users.

Single-precision de-duplication. Duplicates were identified after casting features to float32, which merges some rows that differ only in low-order digits. This removed 22,614 more CIC-IDS2017 rows and 1,900 more CICIOT2023 rows than a float64 comparison would (Section 6). We did not re-run the experiments at float64 precision.

ICVRE rule weights. The rule weights and thresholds were set by design, not learned or validated. For example, ROC was ranked above PR in the CICIOT2023 binary run although attacks outnumber benign flows 27 to 1.

Known dataset errors. We used the original CIC-IDS2017 release, not the corrected versions [2, 11], to stay comparable with prior work; its labelling errors persist in our results.

CONCLUSION AND FUTURE WORK

We presented XVA-Cyber, a reproducible framework that carries cybersecurity data from raw files to statistically validated, explained and provenance-tracked figures and tables, and ICVRE, a transparent rule base that plans which figures an analysis needs. Applied to CIC-IDS2017, CICIOT2023 and UNSW-NB15, the framework showed three things. Tree ensembles outperform linear and neural baselines in every run, and single trees and KNN in most, but cannot be ranked among themselves. Data quality (duplicates and label conflicts) limits achievable scores more than model choice does. For these models, TreeSHAP explanations are cheaper than LIME and, under a deletion test, more faithful to the model. Transfer across datasets on aligned features remains weak. The contribution lies in integrating established techniques with measured evidence of what each component adds, not in a new detector.

Future work will (i) run the expert-rating study for ICVRE and a timed user study of the manual workflow; (ii) repeat all experiments over several seeds and on the full datasets, including corrected CIC-IDS2017 releases; (iii) replace manual

feature alignment with standard NetFlow features [46] to strengthen transfer experiments; and (iv) extend the framework to streaming data and to model-agnostic explainers for deep models.

Funding

[Author to confirm funding source and grant number, or state: This research received no external funding.]

Data Availability Statement

The datasets analysed are publicly available from their providers: CIC-IDS2017 (<https://www.unb.ca/cic/datasets/ids-2017.html>), CICIOT2023 (<https://www.unb.ca/cic/datasets/iotdataset-2023.html>) and UNSW-NB15 (<https://research.unsw.edu.au/projects/unsw-nb15-dataset>). File names and SHA-256 checksums of the exact files used are listed in the reproducibility report. All derived results (metrics, CV scores, test statistics, ICVRE plans) are included in the project archive.

Code Availability Statement

The complete source code, configuration and scripts to reproduce every figure and table are provided with this manuscript [Author to add public repository URL and archived DOI, e.g., Zenodo].

Author Contributions

[Author to complete using CRediT roles, e.g., A.B.: Conceptualization, Methodology, Software, Formal analysis, Writing - original draft; C.D.: Supervision, Writing - review & editing.]

Acknowledgements

[Author to add, or state: Not applicable.]

Conflict of Interest

The authors declare no conflict of interest. [Author to confirm.]

Ethics Approval

Not applicable. The study uses only publicly available, anonymised network-traffic benchmark datasets and involved no human participants.

Informed Consent

Not applicable.

Generative AI Statement

A generative AI assistant (Claude, Anthropic) was used to write and run the analysis code, to search for and verify references, and to draft the manuscript text. All numerical results were produced by executing the released code and are inserted into the text automatically from the result files. The authors reviewed and are responsible for all content. [Author to adapt to the target journal's AI-disclosure policy.]

Supplementary Materials

S1-S6: per-run figures (12 per dataset and task, PNG/SVG/PDF/TIFF), supplementary tables (binary performance, all metrics, binary ablation, full-vs-selected features, tuned hyperparameters, ICVRE suppressions), the literature-evidence table and the reproducibility report.

REFERENCES

1. Sharafaldin, A. Habibi Lashkari, A. A. Ghorbani, "Toward generating a new intrusion detection dataset and intrusion traffic characterization," in Proc. 4th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP), pp. 108-116, 2018. doi: 10.5220/0006639801080116.
2. G. Engelen, V. Rimmer, W. Joosen, "Troubleshooting an intrusion detection dataset: The CICIDS2017 case study," in Proc. IEEE Secur. Privacy Workshops (SPW), pp. 7-12, 2021. doi: 10.1109/SPW53761.2021.00009.
3. E. C. P. Neto, S. Dadkhah, R. Ferreira, A. Zohourian, R. Lu, A. A. Ghorbani, "CICIOT2023: A real-time dataset and benchmark for large-scale attacks in IoT environment," Sensors, vol. 23, no. 13, Art. no. 5941, 2023. doi: 10.3390/s23135941.
4. N. Moustafa, J. Slay, "UNSW-NB15: A comprehensive data set for network intrusion detection systems (UNSW-NB15 network data set)," in Proc. Military Commun. Inf. Syst. Conf. (MilCIS), pp. 1-6, 2015. doi: 10.1109/MilCIS.2015.7348942.
5. N. Moustafa, J. Slay, "The evaluation of network anomaly detection systems: Statistical analysis of the UNSW-NB15 data set and the comparison with the KDD99 data set," Inf. Secur. J.: Glob. Perspect., vol. 25, no. 1-3, pp. 18-31, 2016. doi: 10.1080/19393555.2015.1125974.
6. D. Arp et al., "Pitfalls in machine learning for computer security," Commun. ACM, vol. 67, no. 11, pp. 104-112, 2024. doi: 10.1145/3643456.
7. S. Kapoor, A. Narayanan, "Leakage and the reproducibility crisis in machine-learning-based science," Patterns, vol. 4, no. 9, Art. no. 100804, 2023. doi: 10.1016/j.patter.2023.100804.
8. L. D'hooge, T. Wauters, B. Volckaert, F. De Turck, "Inter-dataset generalization strength of supervised machine learning methods for intrusion detection," J. Inf. Secur. Appl., vol. 54, Art. no. 102564, 2020. doi: 10.1016/j.jisa.2020.102564.
9. M. Cantone, C. Marrocco, A. Bria, "Machine learning in network intrusion detection: A cross-dataset generalization study," IEEE Access, vol. 12, pp. 144489-144508, 2024. doi: 10.1109/ACCESS.2024.3472907.
10. R. Elangovan et al., "Cross-dataset temporal and semantic generalization of intrusion detection models for the future Internet," Future Internet, vol. 18, no. 4, Art. no. 194, 2026. doi: 10.3390/fi18040194.

12. L. Liu, G. Engelen, T. Lynar, D. Essam, W. Joosen, "Error prevalence in NIDS datasets: A case study on CIC-IDS-2017 and CSE-CIC-IDS-2018," in *Proc. IEEE Conf. Commun. Netw. Secur. (CNS)*, pp. 254-262, 2022. doi: 10.1109/CNS56114.2022.9947235.
13. M. Lanvin et al., "Errors in the CICIDS2017 dataset and the significant differences in detection performances it makes," in *Risks and Security of Internet and Systems (CRiSIS 2022)*, LNCS, pp. 18-33, 2023. doi: 10.1007/978-3-031-31108-6_2.
14. A. Rosay, E. Cheval, F. Carlier, P. Leroux, "Network intrusion detection: A comprehensive analysis of CIC-IDS2017," in *Proc. 8th Int. Conf. Inf. Syst. Secur. Privacy (ICISSP)*, pp. 25-36, 2022. doi: 10.5220/0010774000003120.
15. J. Demšar, "Statistical comparisons of classifiers over multiple data sets," *J. Mach. Learn. Res.*, vol. 7, pp. 1-30, 2006.
16. A. Benavoli, G. Corani, F. Mangili, "Should we really use post-hoc tests based on mean-ranks?," *J. Mach. Learn. Res.*, vol. 17, no. 5, pp. 1-10, 2016.
17. S. M. Lundberg, S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 4765-4774, 2017.
18. S. M. Lundberg et al., "From local explanations to global understanding with explainable AI for trees," *Nat. Mach. Intell.*, vol. 2, no. 1, pp. 56-67, 2020. doi: 10.1038/s42256-019-0138-9.
19. M. T. Ribeiro, S. Singh, C. Guestrin, "Why should I trust you?: Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD*, pp. 1135-1144, 2016. doi: 10.1145/2939672.2939778.
20. S. Neupane, J. Ables, W. Anderson, S. Mittal, S. Rahimi, I. Banicescu, M. Seale, "Explainable intrusion detection systems (X-IDS): A survey of current methods, challenges, and opportunities," *IEEE Access*, vol. 10, pp. 112392-112415, 2022. doi: 10.1109/ACCESS.2022.3216617.
21. N. Capuano, G. Fenza, V. Loia, C. Stanzione, "Explainable artificial intelligence in cybersecurity: A survey," *IEEE Access*, vol. 10, pp. 93575-93600, 2022. doi: 10.1109/ACCESS.2022.3204171.
22. N. Moustafa, N. Koroniotis, M. Keshk, A. Y. Zomaya, Z. Tari, "Explainable intrusion detection for cyber defences in the Internet of Things: Opportunities and solutions," *IEEE Commun. Surveys Tuts.*, vol. 25, no. 3, pp. 1775-1807, 2023. doi: 10.1109/COMST.2023.3280465.
23. Arreche, T. R. Guntur, J. W. Roberts, M. Abdallah, "E-XAI: Evaluating black-box explainable AI frameworks for network intrusion detection," *IEEE Access*, vol. 12, pp. 23954-23988, 2024. doi: 10.1109/ACCESS.2024.3365140.
24. P. Hermosilla, S. Berrios, H. Allende-Cid, "Explainable AI for forensic analysis: A comparative study of SHAP and LIME in intrusion detection models," *Appl. Sci.*, vol. 15, no. 13, Art. no. 7329, 2025. doi: 10.3390/app15137329.
25. B. Nugraha, A. V. Jnanashree, T. Bauschert, "A versatile XAI-based framework for efficient and explainable intrusion detection systems," *Ann. Telecommun.*, vol. 80, pp. 1095-1120, 2025. doi: 10.1007/s12243-025-01118-9.
26. G. K. Sandve, A. Nekrutenko, J. Taylor, E. Hovig, "Ten simple rules for reproducible computational research," *PLoS Comput. Biol.*, vol. 9, no. 10, Art. no. e1003285, 2013. doi: 10.1371/journal.pcbi.1003285.
27. N. P. Rougier, M. Droettboom, P. E. Bourne, "Ten simple rules for better figures," *PLoS Comput. Biol.*, vol. 10, no. 9, Art. no. e1003833, 2014. doi: 10.1371/journal.pcbi.1003833.
28. K. Hu, M. A. Bakker, S. Li, T. Kraska, C. Hidalgo, "VizML: A machine learning approach to visualization recommendation," in *Proc. CHI Conf. Hum. Factors Comput. Syst.*, pp. 1-12, 2019. doi: 10.1145/3290605.3300358.
29. D. Moritz et al., "Formalizing visualization design knowledge as constraints: Actionable and extensible models in Draco," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 438-448, 2019. doi: 10.1109/TVCG.2018.2865240.
30. H. Li, Y. Wang, S. Zhang, Y. Song, H. Qu, "KG4Vis: A knowledge graph-based approach for visualization recommendation," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 195-205, 2022. doi: 10.1109/TVCG.2021.3114863.
31. S. Zhang, H. Li, H. Qu, Y. Wang, "AdaVis: Adaptive and explainable visualization recommendation for tabular data," *IEEE Trans. Vis. Comput. Graph.*, vol. 30, no. 9, pp. 5923-5938, 2024. doi: 10.1109/TVCG.2023.3316469.
32. V. Dibia, "LIDA: A tool for automatic generation of grammar-agnostic visualizations and infographics using large language models," in *Proc. 61st Annu. Meet. ACL (Demo)*, pp. 113-126, 2023. doi: 10.18653/v1/2023.acl-demo.11.
33. L. Wang, S. Zhang, Y. Wang, E.-P. Lim, Y. Wang, "LLM4Vis: Explainable visualization recommendation using ChatGPT," in *Proc. EMNLP (Industry Track)*, pp. 675-692, 2023. doi: 10.18653/v1/2023.emnlp-industry.64.
34. C. Yuan et al., "PaperOrchestrator: An LLM-orchestrated multi-agent pipeline for automated end-to-end scientific paper writing," *J. King Saud Univ. Comput. Inf. Sci.*, vol. 38, no. 5, Art. no. 296, 2026. doi: 10.1007/s44443-026-00708-4.
35. J. Pineau et al., "Improving reproducibility in machine learning research (A report from the NeurIPS 2019 reproducibility program)," *J. Mach. Learn. Res.*, vol. 22, no. 164, pp. 1-20, 2021.
36. S. Kapoor et al., "REFORMS: Consensus-based recommendations for machine-learning-based science," *Sci. Adv.*, vol. 10, no. 18, Art. no. eadk3452, 2024. doi: 10.1126/sciadv.adk3452.

46. H. Semmelrock et al., "Reproducibility in machine-learning-based research: Overview, barriers, and drivers," *AI Mag.*, vol. 46, no. 2, Art. no. e70002, 2025. doi: 10.1002/aaai.70002.
47. B. J. Heil et al., "Reproducibility standards for machine learning in the life sciences," *Nat. Methods*, vol. 18, no. 10, pp. 1132-1135, 2021. doi: 10.1038/s41592-021-01256-7.
48. T. Chen, C. Guestrin, "XGBoost: A scalable tree boosting system," in *Proc. 22nd ACM SIGKDD*, pp. 785-794, 2016. doi: 10.1145/2939672.2939785.
49. G. Ke et al., "LightGBM: A highly efficient gradient boosting decision tree," in *Adv. Neural Inf. Process. Syst. (NeurIPS)*, vol. 30, pp. 3146-3154, 2017.
50. U. Ahmed, Z. Jiangbin, A. Almogren, M. Sadiq, A. U. Rehman, M. T. Sadiq, J. Choi, "Hybrid bagging and boosting with SHAP based feature selection for enhanced predictive modeling in intrusion detection systems," *Sci. Rep.*, vol. 14, no. 1, Art. no. 30532, 2024. doi: 10.1038/s41598-024-81151-1.
51. S. A. Almahaqeri et al., "An optimized gradient boosting framework for IoT intrusion detection: A comprehensive evaluation on the CICIoT2023 dataset," *Sci. Rep.*, vol. 16, no. 1, Art. no. 16909, 2026. doi: 10.1038/s41598-026-47399-5.
52. D. D. Bikila, J. Čapek, "Machine learning-based attack detection for the Internet of Things," *Future Gener. Comput. Syst.*, vol. 166, Art. no. 107630, 2025. doi: 10.1016/j.future.2024.107630.
53. S. Yaras, M. Dener, "IoT-based intrusion detection system using new hybrid deep learning algorithm," *Electronics*, vol. 13, no. 6, Art. no. 1053, 2024. doi: 10.3390/electronics13061053.
54. S. Hizal, U. Cavusoglu, D. Akgun, "A novel deep learning-based intrusion detection system for IoT DDoS security," *Internet Things*, vol. 28, Art. no. 101336, 2024. doi: 10.1016/j.iot.2024.101336.
55. B. Sharma, L. Sharma, C. Lal, S. Roy, "Explainable artificial intelligence for intrusion detection in IoT networks: A deep learning based approach," *Expert Syst. Appl.*, vol. 238, Art. no. 121751, 2024. doi: 10.1016/j.eswa.2023.121751.
56. M. Sarhan, S. Layeghy, M. Portmann, "Towards a standard feature set for network intrusion detection system datasets," *Mobile Netw. Appl.*, vol. 27, no. 1, pp. 357-370, 2022. doi: 10.1007/s11036-021-01843-0.
57. M. Sarhan, S. Layeghy, M. Portmann, "Evaluating standard feature sets towards increased generalisability and explainability of ML-based network intrusion detection," *Big Data Res.*, vol. 30, Art. no. 100359, 2022. doi: 10.1016/j.bdr.2022.100359.
58. C. Kılıç, G. Şengül, "SHAP-guided feature selection for cross-dataset generalization in network intrusion detection systems," *IEEE Access*, vol. 14, pp. 90926-90936, 2026. doi: 10.1109/ACCESS.2026.3703481.
59. V. Z. Mohale, I. C. Obagbuwa, "A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhancing transparency and interpretability in cybersecurity," *Front. Artif. Intell.*, vol. 8, Art. no. 1526221, 2025. doi: 10.3389/frai.2025.1526221.
60. S. Al, S. Sagiroglu, "Explainable artificial intelligence models in intrusion detection systems," *Eng. Appl. Artif. Intell.*, vol. 144, Art. no. 110145, 2025. doi: 10.1016/j.engappai.2025.110145.
61. S. Patil et al., "Explainable artificial intelligence for intrusion detection system," *Electronics*, vol. 11, no. 19, Art. no. 3079, 2022. doi: 10.3390/electronics11193079.
62. X. Larriva-Novo, C. Sánchez-Zas, V. A. Villagrà, A. Marín-López, J. Berrocal, "Leveraging explainable artificial intelligence in real-time cyberattack identification: Intrusion detection system approach," *Appl. Sci.*, vol. 13, no. 15, Art. no. 8587, 2023. doi: 10.3390/app13158587.
63. X. Larriva-Novo, L. Pérez Miguel, V. A. Villagrà, M. Álvarez-Campana, C. Sánchez-Zas, Ó. Jover, "Post-hoc categorization based on explainable AI and reinforcement learning for improved intrusion detection," *Appl. Sci.*, vol. 14, no. 24, Art. no. 11511, 2024. doi: 10.3390/app142411511.
64. V. Z. Mohale, I. C. Obagbuwa, "Evaluating machine learning-based intrusion detection systems with explainable AI: Enhancing transparency and interpretability," *Front. Comput. Sci.*, vol. 7, Art. no. 1520741, 2025. doi: 10.3389/fcomp.2025.1520741.
65. X. Chen, M. Liu, Z. Wang, Y. Wang, "Explainable deep learning-based feature selection and intrusion detection method on the Internet of Things," *Sensors*, vol. 24, no. 16, Art. no. 5223, 2024. doi: 10.3390/s24165223.
66. Arreche, T. Guntur, M. Abdallah, "XAI-IDS: Toward proposing an explainable artificial intelligence framework for enhancing network intrusion detection systems," *Appl. Sci.*, vol. 14, no. 10, Art. no. 4170, 2024. doi: 10.3390/app14104170.
67. D. Gaspar, P. Silva, C. Silva, "Explainable AI for intrusion detection systems: LIME and SHAP applicability on multi-layer perceptron," *IEEE Access*, vol. 12, pp. 30164-30175, 2024. doi: 10.1109/ACCESS.2024.3368377.
68. A. E. Muhammad, K.-C. Yow, N. Baćanin-Džakula, M. A. Khan, "L-XAIDS: A LIME-based eXplainable AI framework for intrusion detection systems," *Cluster Comput.*, vol. 28, no. 10, Art. no. 654, 2025. doi: 10.1007/s10586-025-05326-9.
69. H. Shiravi, A. Shiravi, A. A. Ghorbani, "A survey of visualization systems for network security," *IEEE Trans. Vis. Comput. Graph.*, vol. 18, no. 8, pp. 1313-1329, 2012. doi: 10.1109/TVCG.2011.144.
70. J. R. Goodall et al., "Situ: Identifying and explaining suspicious behavior in networks," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 1, pp. 204-214, 2019. doi: 10.1109/TVCG.2018.2865029.
71. M. Rabzelj, C. Bohak, L. Š. Južnič, A. Kos, U. Sedlar, "Cyberattack graph modeling for visual analytics," *IEEE Access*, vol. 11, pp. 86910-86944, 2023. doi: 10.1109/ACCESS.2023.3304640.
72. J. Tang et al., "Towards enhancing inter-domain routing security with visualization and visual analytics," *IEEE Trans. Big Data*, vol.

74. 11, no. 3, pp. 1508-1527, 2025. doi: 10.1109/TBDATA.2024.3481899.
75. L. Zheng, G. Yu, Y. Zheng, "An interactive visualization system for network security data," *J. Signal Process. Syst.*, vol. 95, no. 2-3, pp. 393-405, 2023. doi: 10.1007/s11265-023-01836-0.
77. S. Chavan, G. Pappas, "From detection to decision: Transforming cybersecurity with deep learning and visual analytics," *AI*, vol. 6, no. 9, Art. no. 214, 2025. doi: 10.3390/ai6090214.
78. F. Hohman, M. Kahng, R. Pienta, D. H. Chau, "Visual analytics in deep learning: An interrogative survey for the next frontiers," *IEEE Trans. Vis. Comput. Graph.*, vol. 25, no. 8, pp. 2674-2693, 2019. doi: 10.1109/TVCG.2018.2843369.
79. X. Qian et al., "Learning to recommend visualizations from data," in *Proc. 27th ACM SIGKDD*, pp. 1359-1369, 2021. doi: 10.1145/3447548.3467224.
80. A. Wu et al., "MultiVision: Designing analytical dashboards with deep learning based recommendation," *IEEE Trans. Vis. Comput. Graph.*, vol. 28, no. 1, pp. 162-172, 2022. doi: 10.1109/TVCG.2021.3114826.
81. S. Zhu, G. Sun, Q. Jiang, M. Zha, R. Liang, "A survey on automatic infographics and visualization recommendations," *Vis. Informatics*, vol. 4, no. 3, pp. 24-40, 2020. doi: 10.1016/j.visinf.2020.07.002.
83. M. D. Wilkinson et al., "The FAIR guiding principles for scientific data management and stewardship," *Sci. Data*, vol. 3, Art. no. 160018, 2016. doi: 10.1038/sdata.2016.18.
84. A. Paleyes, R.-G. Urma, N. D. Lawrence, "Challenges in deploying machine learning: A survey of case studies," *ACM Comput. Surv.*, vol. 55, no. 6, pp. 1-29, 2022. doi: 10.1145/3533378.
85. R. Gerum, "pylustrator: Code generation for reproducible figures for publication," *J. Open Source Softw.*, vol. 5, no. 51, Art. no. 1989, 2020. doi: 10.21105/joss.01989.
86. W. Liang et al., "Quantifying large language model usage in scientific papers," *Nat. Hum. Behav.*, vol. 9, no. 12, pp. 2599-2609, 2025. doi: 10.1038/s41562-025-02273-8.
87. D. Kobak, R. González-Márquez, E.-Á. Horvát, J. Lause, "Delving into LLM-assisted writing in biomedical publications through excess vocabulary," *Sci. Adv.*, vol. 11, no. 27, Art. no. eadt3813, 2025. doi: 10.1126/sciadv.adt3813.
88. C. A. Gao et al., "Comparing scientific abstracts generated by ChatGPT to real abstracts with detectors and blinded human reviewers," *npj Digit. Med.*, vol. 6, no. 1, Art. no. 75, 2023. doi: 10.1038/s41746-023-00819-6.
89. B. D. Lund et al., "ChatGPT and a new academic reality: Artificial Intelligence-written research papers and the ethics of the large language models in scholarly publishing," *J. Assoc. Inf. Sci. Technol.*, vol. 74, no. 5, pp. 570-581, 2023. doi: 10.1002/asi.24750.
90. N. V. Chawla, K. W. Bowyer, L. O. Hall, W. P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," *J. Artif. Intell. Res.*, vol. 16, pp. 321-357, 2002. doi: 10.1613/jair.953.
92. M. Friedman, "The use of ranks to avoid the assumption of normality implicit in the analysis of variance," *J. Amer. Statist. Assoc.*, vol. 32, no. 200, pp. 675-701, 1937. doi: 10.1080/01621459.1937.10503522.
93. F. Wilcoxon, "Individual comparisons by ranking methods," *Biometrics Bull.*, vol. 1, no. 6, pp. 80-83, 1945. doi: 10.2307/3001968.
94. S. García, A. Fernández, J. Luengo, F. Herrera, "Advanced nonparametric tests for multiple comparisons in the design of experiments in computational intelligence and data mining: Experimental analysis of power," *Inf. Sci.*, vol. 180, no. 10, pp. 2044-2064, 2010. doi: 10.1016/j.ins.2009.12.010.
95. F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, pp. 2825-2830, 2011.
96. L. McInnes, J. Healy, N. Saul, L. Großberger, "UMAP: Uniform manifold approximation and projection," *J. Open Source Softw.*, vol. 3, no. 29, Art. no. 861, 2018. doi: 10.21105/joss.00861.
97. J. D. Hunter, "Matplotlib: A 2D graphics environment," *Comput. Sci. Eng.*, vol. 9, no. 3, pp. 90-95, 2007. doi: 10.1109/MCSE.2007.55.