STATISTICS, OPTIMIZATION AND INFORMATION COMPUTING
Stat., Optim. Inf. Comput., Vol. **x**, Month 2026, pp 1–16.
Published online in International Academic Press (www.IAPress.org)

Cross-Model SHAP Agreement in Network Intrusion Detection: Consensus, Divergence, and Dataset Artifacts

Mohammed BERHILI*, Omar CHAIEB, Mohammed BENABDELLAH

SIOSI Research Team, LIPIO Laboratory, Faculty of Sciences of Oujda, Mohammed First University, Oujda 60000, Morocco

Abstract **Context.** Tree-based ensemble methods consistently rank among the best-performing classifiers for network intrusion detection systems (IDS), and explainability tools such as SHAP are increasingly used to interpret their predictions. However, most studies apply SHAP to a single model in isolation, leaving open the question of whether different models actually agree on the features that drive their decisions. **Objective.** We address this gap by quantifying cross-model SHAP agreement on UNSW-NB15 and re-framing it as a *diagnostic* tool for detecting dataset artifacts that single-model interpretation cannot surface. **Method.** This paper applies SHAP TreeExplainer to XGBoost, LightGBM, and RandomForest trained on the official UNSW-NB15 partition and measures pairwise inter-model consensus using Spearman rank correlation, Kendall τ, and top-k feature overlap, with 95% bootstrap confidence intervals ($B = 1{,}000$) on every agreement metric. **Results.** XGBoost and LightGBM exhibit strong agreement ($\rho = 0.861$, 100% top-5 overlap), while both boosting models agree only moderately with RandomForest ($\rho \approx 0.65$). All three models rank `sttl` (source-to-destination TTL) as the most important feature by a wide margin, a finding that reveals a shared dependence on a network-layer artifact of the UNSW-NB15 lab topology rather than a generalizable attack signature. An ablation experiment confirms that removing `sttl` does not degrade accuracy ($\Delta \leq 0.04\%$ for all three models), because the artifact is distributed across a cluster of correlated TTL features. Adding a multi-layer perceptron separates two effects that accuracy and single-model interpretation conflate: rank agreement halves outside the tree family ($\rho = 0.480$–0.536), yet three of the MLP's five top-ranked features remain TTL-cluster members. Consensus is therefore partly a property of the model family; the artifact is a property of the dataset. Permutation importance, an attribution method independent of SHAP, identifies the same cluster (permuting it costs 22–28 accuracy points against 1.4 for random features), and of the 56,157 flows with `sttl` $= 31$ not one is an attack. **Implication.** We argue that cross-model SHAP agreement should become a standard diagnostic in IDS evaluation: consensus on a small feature set is not evidence of robustness if the consensus features lack semantic validity; pairwise agreement metrics together with semantic inspection of consensus features should be reported alongside accuracy in every SHAP-based IDS study.

Keywords intrusion detection system; SHAP; feature importance; XGBoost; LightGBM; RandomForest; explainability; UNSW-NB15; model agreement

DOI: 10.19139/soic-2310-5070-4677

1. Introduction

Network intrusion detection benchmarks now routinely report accuracies above 97% on datasets like UNSW-NB15 [1] and NSL-KDD [2], driven largely by tree-based ensembles. XGBoost [3], LightGBM [4], and RandomForest [5] consistently rank among the best-performing classifiers on structured network traffic, often ahead of deeper architectures [11]. High accuracy on a benchmark test set is nonetheless a weak guarantee: a model can latch onto statistical regularities that are specific to one dataset's collection environment without ever learning anything about what makes traffic malicious, a failure mode that the broader machine learning literature now calls *shortcut learning* [20].

*Correspondence to: Mohammed BERHILI (Email: mohammed.berhili@outlook.com). SIOSI Research Team, LIPIO Laboratory, Faculty of Sciences of Oujda, Mohammed First University, Oujda 60000, Morocco.

ISSN 2310-5070 (online) ISSN 2311-004X (print)

This gap between accuracy and understanding is where interpretability comes in. SHAP (SHapley Additive exPlanations) [6] attributes each prediction to individual features using Shapley values from cooperative game theory, with formal guarantees that simpler importance scores do not satisfy. The TreeExplainer extension [7] makes this tractable for tree ensembles by traversing the tree structure directly, yielding exact values without approximation.

Most existing work applies SHAP to a single model in isolation, asking *what does this model learn?* [8, 9]. A more fundamental question for IDS deployment is whether different models, trained on the same data with different inductive biases, reach the same conclusions about feature importance. If XGBoost, LightGBM, and RandomForest all agree on the top features, that consensus is evidence that those features carry genuine signal. If they disagree, the reported accuracy conceals a fragility: each model may be exploiting a different dataset artifact, none of which generalise to real traffic.

This paper quantifies cross-model SHAP agreement on UNSW-NB15 and interprets what the consensus reveals about the features being learned.

Novelty statement. To the best of our knowledge, this work is the first to (i) measure pairwise inter-model SHAP agreement on the UNSW-NB15 benchmark with three complementary statistical metrics (Spearman ρ, Kendall τ, and top-k overlap) accompanied by bootstrap confidence intervals, (ii) demonstrate that the dominant consensus feature is a network-layer artifact of the dataset rather than a generalisable attack signature, and (iii) re-frame cross-model SHAP agreement from a *validation* tool (consensus = validation) into a *diagnostic* tool (consensus = warrants semantic inspection). Earlier work that applies SHAP to IDS, such as [8, 9], considers only a single model, while comparative studies such as [10] report aggregate accuracy similarities without inspecting the features themselves.

Our contributions are as follows:

1. We compute SHAP values for XGBoost, LightGBM, and RandomForest on the UNSW-NB15 official test partition using TreeExplainer, and rank all 42 features by mean absolute SHAP value for each model.
2. We measure pairwise inter-model agreement using Spearman rank correlation, Kendall τ, and top-k feature overlap for $k = 1$ to 42, providing the first systematic quantification of this kind on UNSW-NB15, and we report 95% bootstrap confidence intervals for each agreement metric.
3. We interpret the results from a security standpoint, showing that the dominant consensus feature (`sttl`, source-to-destination TTL) is a network-layer artifact of the UNSW-NB15 lab topology rather than a semantically meaningful attack indicator, and we discuss the implications for IDS generalisation.
4. We propose that pairwise cross-model SHAP agreement metrics, complemented by an explicit semantic inspection of the consensus features, become a standard reporting practice in SHAP-based IDS studies.
5. We separate agreement attributable to model family from agreement attributable to the dataset. Adding a multi-layer perceptron halves the rank correlation with the boosting models, yet three of its five top-ranked features remain TTL-cluster members: the consensus is partly a tree phenomenon, the artifact is not.

Motivation. Intrusion-detection benchmarks are routinely reported at accuracies between 87% and 99%, and SHAP is increasingly used to explain the models that reach them. Neither number tells a reader whether a model has learned attack behaviour or a property of the testbed on which the data were captured. Accuracy cannot make that distinction by construction, and single-model SHAP analysis (the dominant practice) cannot either, because a plausible-looking explanation of one model is equally consistent with both cases. What is missing is an instrument that uses the explanations themselves as evidence about the data. This paper proposes cross-model agreement as that instrument, and demonstrates on UNSW-NB15 that it surfaces an artifact which accuracy alone rewards and single-model interpretation alone conceals.

The remainder of the paper is organised as follows. Section 2 reviews related work on SHAP-based IDS interpretability. Section 3 describes the dataset, models, and agreement metrics. Section 4 presents the SHAP rankings, Spearman matrix, and top-k overlap curves. Section 5 interprets the findings and their deployment implications. Section 6 concludes and outlines future directions.

2. Related Work

As ML-based IDS models have proliferated, so has interest in understanding what they actually learn. SHAP [6] has become the standard attribution tool for this: it decomposes each prediction into per-feature contributions grounded in Shapley values, a concept from cooperative game theory. Properties such as local accuracy and consistency (which simpler permutation or gradient-based measures do not satisfy) follow from this formulation directly. The TreeExplainer extension [7] makes exact SHAP computation tractable for tree ensembles by traversing the tree structure rather than approximating it, yielding precise values at a cost that scales polynomially with tree depth.

The application of SHAP to intrusion detection has gained traction in recent years. Younisse et al. [8] applied SHAP to explain the decisions of a CNN-based IDS, demonstrating that SHAP can surface the features driving individual predictions and providing a global importance ranking for network traffic classification. Sivamohan and Sridhar [9] combined SHAP and LIME to interpret a Bi-LSTM model on NSL-KDD, emphasising that interpretability is as important as accuracy for security analysts who must act on model decisions. Mohale and Obagbuwa [14] apply SHAP, LIME, and ELI5 to XGBoost, CatBoost, and Decision Trees on UNSW-NB15 and independently identify `sttl` as a critical malicious-activity indicator, but stop short of asking whether different models would all converge on it. A recent systematic review of XAI in IDS [15] confirms that SHAP and LIME dominate the literature but highlights the absence of standardised evaluation metrics for cross-method or cross-model consistency. Both studies, like most of this line of work, apply SHAP to a single model: they ask *what does this model learn?* rather than *do different models agree on what matters?*

The question of inter-model agreement is taken up, indirectly, by Corea et al. [10], who apply Occlusion Sensitivity to seven classifiers on UNSW-NB15 and report that most models concentrate on fewer than three features to achieve 90% accuracy. This finding hints at a shared learning pattern, but the study neither uses a theoretically grounded attribution method nor quantifies the degree of agreement between model pairs. No Spearman rank correlation, top-k overlap, or any other agreement metric is reported. Subasi et al. [16] provide a complementary critique, showing empirically that feature-importance rankings produced by different explanation methods on the same IDS model are often inconsistent, and arguing that current XAI evaluation in IDS is methodologically fragile. Our work addresses a related but distinct gap: rather than comparing explanation methods on one model, we fix the explanation method (SHAP TreeExplainer) and measure consistency *across models*.

Our work fills this gap. We apply SHAP TreeExplainer to XGBoost, LightGBM, and RandomForest (the three tree-based classifiers that consistently dominate IDS benchmarks [11]) and measure inter-model consensus using Spearman rank correlation, Kendall τ, and top-k feature overlap. Beyond reporting agreement magnitudes, the comparison also identifies *which features* the models converge on, a step that makes it possible to judge whether the shared signal is a genuine attack pattern or a peculiarity of the benchmark.

3. Methodology

3.1. Dataset and Preprocessing

We use the UNSW-NB15 dataset [1], generated at the Australian Centre for Cyber Security using the IXIA PerfectStorm tool to produce realistic modern network traffic. The dataset contains 257,673 records across nine attack categories (backdoors, analysis, exploits, fuzzers, reconnaissance, shellcode, worms, DoS, and generic) alongside normal traffic, described by 42 features covering flow statistics, content features, time features, and connection features. A full description of every feature, including its physical meaning and unit, is given in the original dataset paper [1]. We use the official train/test partition provided by the dataset authors: 175,341 training samples and 82,332 test samples. This split is fixed and reproducible; using it avoids the inflated accuracy that results from custom random splits on this dataset [12].

Preprocessing follows a strict no-leakage pipeline. Categorical features (`proto`, `service`, `state`) are label-encoded using encoders fitted on the training set only. Label encoding (i.e. mapping each categorical level to an integer) is preferred here over one-hot encoding and over target encoding for three reasons. (i) The three tree-based classifiers we use (XGBoost, LightGBM, RandomForest) split on numerical thresholds and are insensitive

to the absolute ordering of integer codes; a feature whose integer values are arbitrarily permuted yields the same partitioning behaviour, so the artificial ordinality introduced by label encoding has no effect on the learned trees. (ii) One-hot encoding the `service` feature alone (which has > 30 levels in UNSW-NB15) would inflate the feature count from 42 to over 70, dilute each SHAP attribution across many sparse indicator columns, and obscure the cross-model comparison that is the central concern of this paper. (iii) Target encoding would leak label information into the feature representation, which is incompatible with our strict no-leakage protocol. Label encoding therefore preserves both the interpretability of per-feature SHAP scores and the structural integrity of the train/test partition. Numerical features are normalised with MinMaxScaler, also fitted on the training set and applied to the test set without re-fitting. Missing values are imputed with training-set medians (numerical) and modes (categorical). The identifier columns `id` and `attack_cat` are dropped; the binary `label` column (0 = normal, 1 = attack) serves as the classification target. To address the class imbalance in the training set (56% attack, 44% normal after removing identifiers), we apply SMOTE [13] on the training partition only, yielding 238,682 balanced training samples. The test set is never oversampled.

3.2. Models

We train three tree-based classifiers, using the same hyperparameters as the benchmark in [12] to ensure reproducibility:

- **XGBoost** [3]: 2,000 trees, `max_depth`=15, `learning_rate`=0.02, `subsample`=0.8, `colsample_bytree`=0.8, `random_state`=42.
- **LightGBM** [4]: 2,000 trees, `max_depth`=18, `learning_rate`=0.02, leaf-wise growth, `subsample`=0.8, `random_state`=42.
- **RandomForest** [5]: 500 trees, no depth limit, Gini impurity, `random_state`=42.

All models are trained on the SMOTE-balanced training set and evaluated on the official test set. The binary classification accuracies obtained are 87.35% (XGBoost), 88.12% (LightGBM), and 88.28% (RandomForest), consistent with results obtained under proper evaluation protocols on this dataset.

3.3. SHAP Feature Importance

We compute SHAP values using `shap.TreeExplainer` [7] with `feature_perturbation="tree_path_dependent"`, which traverses the tree structure directly without requiring a background dataset and provides exact Shapley values for tree ensembles. To keep computation tractable (TreeExplainer for RandomForest with 500 unlimited-depth trees scales poorly beyond a few thousand samples), we apply it to a stratified random subsample of 200 test samples (90 normal, 110 attack), drawn with `random_state`=42.

To verify that this subsample size yields stable rankings, we measured the Spearman rank correlation between the mean |SHAP| rankings obtained at each size $n \in \{20, 40, \ldots, 175\}$ and the reference ranking at $n = 200$ (Figure 1). All three models reach $\rho \geq 0.985$ at $n = 20$ and $\rho \geq 0.993$ at $n = 60$. The 200-sample choice is therefore conservative: rankings stabilise well before this threshold for all three models on this 42-feature dataset.

For each model $m \in \{\text{XGBoost}, \text{LightGBM}, \text{RandomForest}\}$, we obtain a SHAP value matrix $\Phi^{(m)} \in \mathbb{R}^{200\times 42}$, where $\Phi^{(m)}_{i,j}$ is the SHAP value of feature j for sample i under model m. The global feature importance score for feature j under model m is:

$$s_j^{(m)} = \frac{1}{200} \sum_{i=1}^{200} \left| \Phi_{i,j}^{(m)} \right| \tag{1}$$

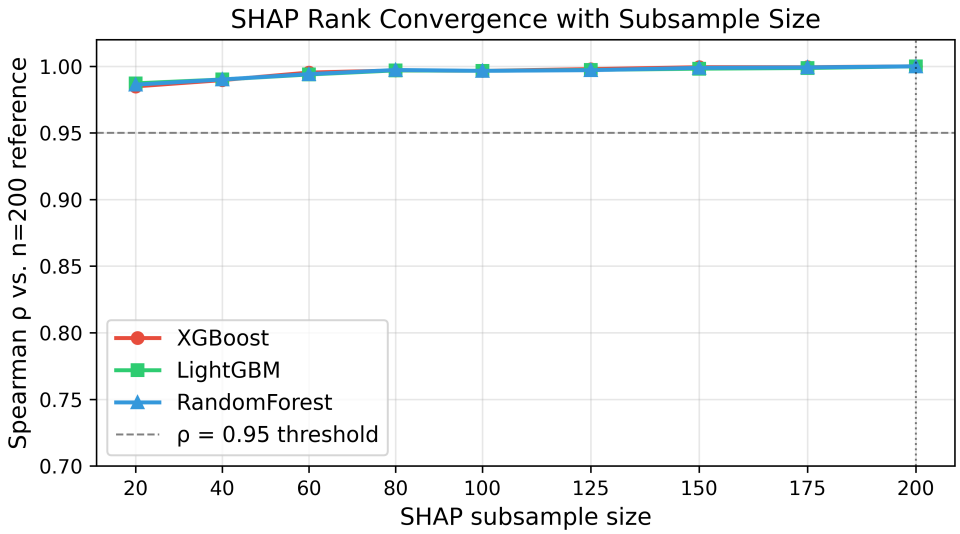


Figure 1. Spearman ρ between SHAP feature rankings at subsample size n and the reference ranking at $n = 200$. All three models reach $\rho \geq 0.985$ at $n = 20$, confirming that the 200-sample choice is conservative.

The feature ranking $r^{(m)}$ is obtained by sorting features in descending order of $s_j^{(m)}$.

3.4. Agreement Metrics

Three metrics are used to quantify consensus between model pairs.

Spearman rank correlation. For each pair (m_1, m_2), we compute ρ between the two importance score vectors $(s_1^{(m_1)}, \ldots, s_{42}^{(m_1)})$ and $(s_1^{(m_2)}, \ldots, s_{42}^{(m_2)})$. $\rho = 1$ means the two models rank features identically; $\rho = 0$ means no monotonic relationship. The three pairwise values populate a 3×3 symmetric matrix.

Kendall τ. Computed alongside Spearman ρ for each pair. Unlike ρ, Kendall τ counts concordant minus discordant feature pairs directly, making it less sensitive to outlier ranks and providing a cross-check on the Spearman result.

Top-k overlap. For $k = 1, \ldots, 42$, the overlap between m_1 and m_2 is:

$$\text{overlap}_k(m_1, m_2) = \frac{\left|\mathcal{T}_k^{(m_1)} \cap \mathcal{T}_k^{(m_2)}\right|}{k} \times 100\% \tag{2}$$

where $\mathcal{T}_k^{(m)}$ is the set of the k top-ranked features under model m. This is a practitioner-oriented metric: if a security analyst retains only the top-k SHAP features for a downstream task, it tells them how often two models would hand them the same candidate set.

Bootstrap confidence intervals. To quantify the sampling variability of the agreement metrics, we apply a non-parametric percentile bootstrap on the 200-sample SHAP matrix. For each iteration $b = 1, \ldots, B$ with $B = 1{,}000$ and a fixed seed, we draw a sample of 200 row indices with replacement from $\{1, \ldots, 200\}$, recompute the per-feature mean |SHAP| score $s_j^{(m)}$ for each model, and re-evaluate ρ, τ, and overlap_k on the resampled scores. The 95% confidence interval for each metric is the 2.5^{th}–97.5^{th} percentile of its B-vector. This procedure resamples *instances* (over which the mean |SHAP| is computed), not features, and therefore quantifies the uncertainty due to the finite explanatory subsample.

4. Results

4.1. Model Accuracy and Classification Metrics

All three models are trained on the SMOTE-balanced training set (238,682 samples) and evaluated on the official UNSW-NB15 test set (82,332 samples). Test accuracies are 87.35% for XGBoost, 88.12% for LightGBM, and 88.28% for RandomForest. These figures are lower than the 97%+ commonly reported in the literature, where custom random splits from the same pool inflate the test score; using the official partition avoids this bias and yields reproducible, comparable results [12]. The three models are comparable in predictive performance, making a feature importance comparison meaningful: differences in SHAP rankings reflect differences in learned representations, not differences in model quality.

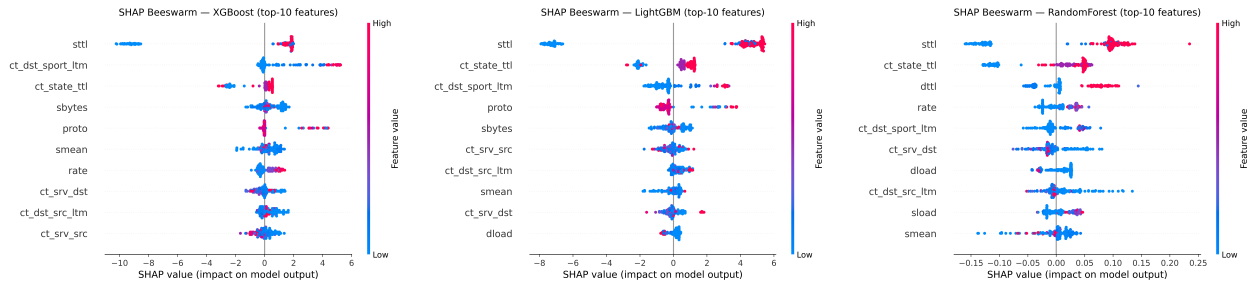
Beyond accuracy, Table 1 reports precision, recall, F_1, and ROC-AUC on the official test set. All three classifiers exhibit very high recall (97.9–98.2%) and ROC-AUC (above 98%), with more modest precision (82.4–83.4%): on this dataset the models behave as sensitive but somewhat over-eager attack detectors, raising approximately $9{,}000$ false alarms on $37{,}000$ normal flows in exchange for missing fewer than $1{,}000$ attacks out of $45{,}332$. The three models again sit within one percentage point of each other on every metric, reinforcing the observation that they are comparable in predictive quality and that differences in SHAP rankings reflect representation, not performance.

Table 1. Classification metrics on the official UNSW-NB15 test set (binary, attack vs. normal). TN/FP/FN/TP are confusion-matrix counts on the 82,332 test samples.

Model	Accuracy	Precision	Recall	F_1	ROC-AUC
XGBoost	0.8735	0.8241	0.9795	0.8951	0.9816
LightGBM	0.8812	0.8325	0.9818	0.9010	0.9849
RandomForest	0.8828	0.8342	0.9822	0.9022	0.9818

4.2. SHAP Feature Rankings

Figure 2 shows the SHAP beeswarm plots for each model (top-10 features by mean |SHAP|). Each dot represents one test sample; its horizontal position indicates the SHAP value (positive = pushes toward attack prediction) and its colour indicates the raw feature value (red = high, blue = low). The figure groups three sub-panels (a)–(c) into a single Figure 2. Table 2 (below) reports the top-10 features by mean |SHAP| across all three models in a single comparison table (one rank column, one column per model).



(a) XGBoost (b) LightGBM (c) RandomForest

Figure 2. SHAP beeswarm plots for the three classifiers. Top-10 features by mean |SHAP|, rendered at 600 dpi for print legibility. Each dot represents one test sample; colour encodes the raw feature value (red: high, blue: low). Panels (a)–(c) are sub-figures of a single Figure 2.

Table 2. Top-10 features by mean |SHAP| across the three models, presented as a single comparison table with one rank column and one feature column per model. Features appearing in all three top-10 lists are marked with †.

Rank	XGBoost	LightGBM	RandomForest
1	sttl†	sttl†	sttl†
2	ct_dst_sport_ltm†	ct_state_ttl†	ct_state_ttl†
3	ct_state_ttl†	ct_dst_sport_ltm†	dttl
4	sbytes	proto	rate
5	proto	sbytes	ct_dst_sport_ltm†
6	smean†	ct_srv_src	ct_srv_dst†
7	rate	ct_dst_src_ltm†	dload
8	ct_srv_dst†	smean†	ct_dst_src_ltm†
9	ct_dst_src_ltm†	ct_srv_dst†	sload
10	ct_srv_src	dload	smean†

The most striking observation is the dominance of sttl (source-to-destination TTL): it ranks first for all three models, with a mean |SHAP| value of 3.62 (XGBoost), 5.35 (LightGBM), and 0.108 (RandomForest). The gap between sttl and the second-ranked feature is large for all three classifiers, suggesting that a single feature drives a disproportionate share of each model's predictions. Six features appear in all three top-10 lists (marked †): sttl, ct_state_ttl, ct_dst_sport_ltm, smean, ct_srv_dst, and ct_dst_src_ltm.

Figure 3 shows the mean |SHAP| values for the union of the three models' top-10 features, side by side.

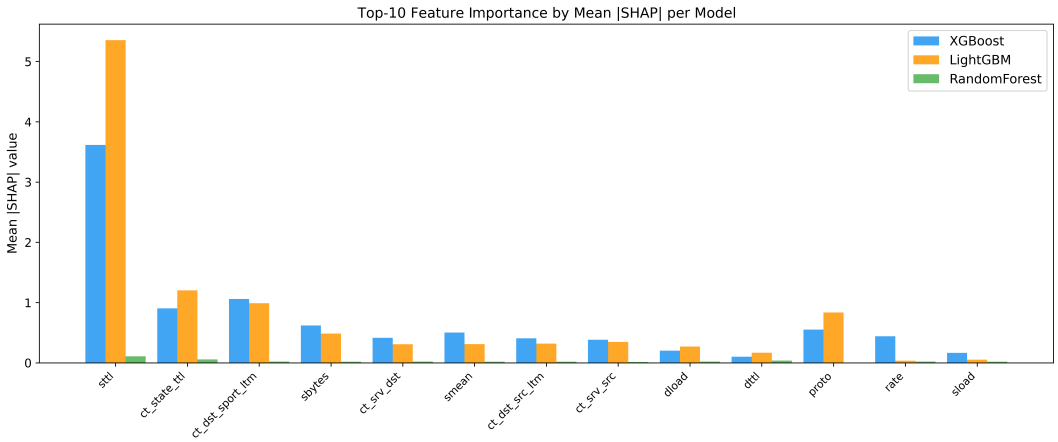


Figure 3. Mean |SHAP| values for the top-10 feature union across all three models.

4.3. Inter-Model Agreement

Spearman and Kendall correlations. Table 3 reports the pairwise Spearman rank correlation coefficients and Kendall τ values, together with their 95% bootstrap confidence intervals ($B = 1{,}000$ resamples). The Spearman matrix is visualised in Figure 4, and Figure 5 shows the corresponding 95% bootstrap intervals.

Table 3. Pairwise Spearman rank correlation (ρ) and Kendall τ between model feature importance rankings. All p-values are below 10^{-5}. 95% bootstrap CIs (percentile method, $B = 1{,}000$) are reported in brackets.

Model Pair	**Spearman ρ**	**95% CI**	**Kendall τ**	**95% CI**
XGBoost vs LightGBM	0.861	[0.843, 0.873]	0.712	[0.691, 0.731]
XGBoost vs RandomForest	0.662	[0.613, 0.677]	0.487	[0.461, 0.515]
LightGBM vs RandomForest	0.647	[0.599, 0.680]	0.496	[0.452, 0.524]

The confidence intervals confirm the qualitative pattern: the XGBoost–LightGBM agreement (Spearman ρ CI $\subset [0.84, 0.88]$) is clearly separated from the boosting–RandomForest agreement (CIs $\subset [0.59, 0.68]$), with no overlap between the two regimes. The narrow widths (≤ 0.03 for ρ, ≤ 0.04 for τ) indicate that the 200-sample SHAP subset yields stable agreement estimates.

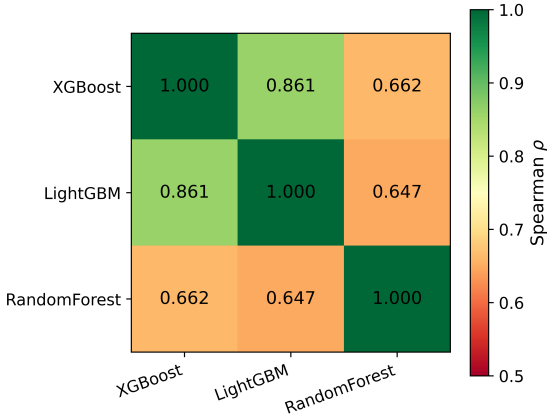


Figure 4. Spearman rank correlation matrix between model pairs. Colour scale: green = high agreement, red = low agreement.

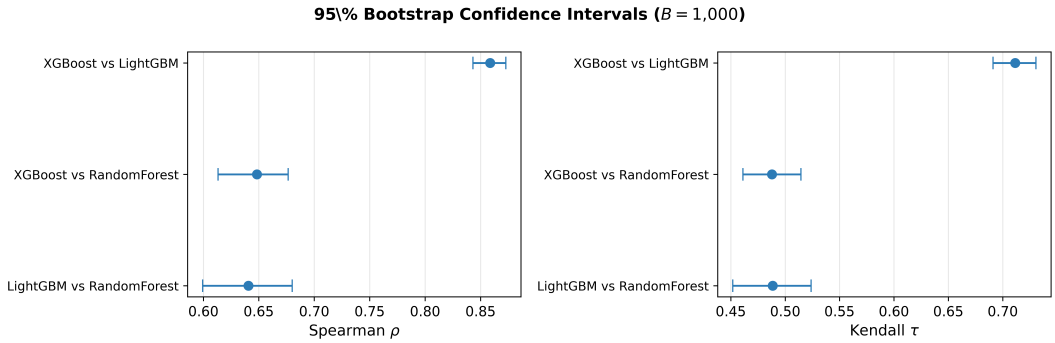


Figure 5. 95% bootstrap confidence intervals ($B = 1{,}000$ percentile resamples) for the pairwise Spearman ρ (left) and Kendall τ (right). The boosting–boosting pair sits clearly above the boosting–RandomForest pairs with non-overlapping intervals.

XGBoost and LightGBM exhibit strong agreement ($\rho = 0.861$, $\tau = 0.712$), consistent with their shared gradient boosting inductive bias. Both boosting models show moderate agreement with RandomForest ($\rho \approx 0.65$), which employs bagging and random feature subsampling rather than sequential residual fitting.

Top-k overlap. Figure 6 plots the top-k overlap curves for $k = 1$ to 42.

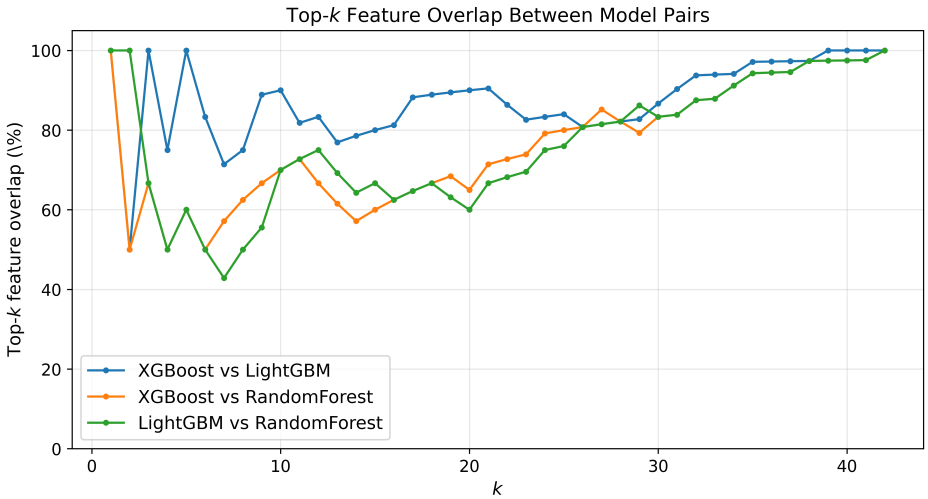


Figure 6. Top-k feature overlap (%) between model pairs for $k = 1$ to 42.

Table 4. Top-k feature overlap (%) at selected values of k, with 95% bootstrap CIs in brackets ($B = 1{,}000$).

Model Pair	$k = 5$	$k = 10$	$k = 15$	$k = 20$
XGBoost vs LightGBM	100.0% [80, 100]	90.0% [80, 90]	80.0% [73, 87]	90.0% [80, 90]
XGBoost vs RandomForest	60.0% [40, 60]	70.0% [60, 80]	60.0% [53, 67]	65.0% [65, 70]
LightGBM vs RandomForest	60.0% [40, 60]	70.0% [50, 80]	66.7% [60, 67]	60.0% [60, 70]

XGBoost and LightGBM achieve 100% top-5 overlap: their five most important features are identical. This overlap decreases gradually as k increases but remains above 80% up to $k = 15$. By contrast, both boosting models share only 60% of their top-5 features with RandomForest, rising modestly to 70% at $k = 10$ before declining again. The boosting–RF overlap curves remain consistently lower and flatter than the XGBoost–LightGBM curve across the full range of k.

4.4. Cross-Architecture Agreement

The three classifiers analysed so far all belong to the tree-ensemble family. Because decision trees share an inductive bias (axis-aligned splits on numerical thresholds), part of the agreement reported in Section 4 may be attributable to model family rather than to the data. To separate the two effects, we add a multi-layer perceptron (MLP), trained on the same SMOTE-balanced partition, and compute its SHAP values under the same protocol.

Table 5 reports the pairwise agreement of the resulting four-model portfolio. The two MLP–boosting pairs are the weakest in the entire study.

Table 5. Pairwise SHAP rank agreement including the MLP. The two weakest pairs are the MLP–boosting pairs.

Model Pair	Spearman ρ	Kendall τ	Top-5 overlap
XGBoost vs LightGBM	0.861	0.712	100%
MLP vs RandomForest	0.649	0.509	80%
XGBoost vs RandomForest	0.662	0.487	60%
LightGBM vs RandomForest	0.647	0.496	60%
MLP vs LightGBM	0.536	0.423	60%
MLP vs XGBoost	0.480	0.395	60%

Global rank agreement therefore does depend on model family: leaving the tree ensembles halves the Spearman coefficient relative to the boosting–boosting pair ($0.861 \to 0.480$). Interpreted alone, this would weaken the case for treating consensus as evidence about the data.

The feature-level picture is different. The MLP's five highest-ranked features are `dttl` (rank 1), `swin`, `sttl` (rank 3), `ct_dst_sport_ltm`, and `ct_state_ttl` (rank 5): three of the five belong to the TTL cluster. A model that shares no architectural component with the ensembles, and that disagrees with them about the ordering of most features, nevertheless converges on the same network-layer quantities.

We therefore distinguish two claims that were conflated in earlier formulations of this work. Rank agreement is partly a property of the model family and should not be read as evidence about the dataset. The dominance of the TTL cluster is not: it survives a change of hypothesis class, and is a property of UNSW-NB15 itself.

4.5. Beyond SHAP: Permutation Importance

A consensus measured with a single attribution method could in principle be a property of that method. We therefore repeat the analysis with permutation feature importance, which shares no machinery with SHAP: it computes no Shapley values and makes no additivity assumption. We permute the TTL cluster on the official test partition and, as a placebo control, random triplets of features (five repetitions).

Table 6. Accuracy loss under permutation of the TTL cluster versus random triplets (mean $\pm$ s.d. over five repetitions), official UNSW-NB15 test set.

Model	TTL cluster permuted	3 random features permuted
XGBoost	-26.94 pts	-1.36 ± 2.11 pts
LightGBM	-27.88 pts	-1.61 ± 1.92 pts
RandomForest	-22.67 pts	-0.78 ± 2.16 pts

The effect exceeds the placebo band by more than an order of magnitude for every model. A second, independent attribution method identifies the same cluster as decisive, which rules out the interpretation that the consensus is an artifact of SHAP. This is consistent with evidence from this journal: Aljarrah et al. [17] report correlations above 0.84 across Integrated Gradients, SHAP, and LIME on a related security-detection task.

4.6. Extended Ablation: the TTL Cluster

Section 4 removed only the single dominant feature. Since we argue that the artifact is carried by several correlated variables, we now ablate the whole TTL cluster (`sttl`, `dttl`, `ct_state_ttl`), retrain from scratch, and compare against a control in which three *random* features are removed instead.

Table 7. Test accuracy after retraining without the TTL cluster, against a random-removal control (mean over five repetitions).

Model	All 42 features	TTL cluster removed	3 random removed
XGBoost	0.8735	0.8721	0.8756
LightGBM	0.8812	0.8790	0.8849
RandomForest	0.8828	0.8821	0.8848

Removing the entire cluster costs between 0.07 and 0.22 accuracy points, within the run-to-run variation of the random-removal control. The redundancy therefore extends beyond the three TTL features: once they are unavailable, the models recover equivalent discriminative information from other variables.

Read together with Section 4.5, this yields the mechanistic statement of the paper. The models *depend* on the TTL cluster (permuting it costs 22–28 accuracy points), but are not *constrained* to it, since removing it before training costs nothing. Dependence without constraint is the signature of a redundantly encoded dataset artifact, and it is precisely why single-feature ablation, which we reported in earlier versions of this work, understates the phenomenon.

4.7. Robustness of the Rankings

SHAP sample size and random seed. The rankings above are computed on a 200-instance stratified subsample of the 82,332-instance test set. To verify that this budget is sufficient, we recomputed the rankings for sample sizes $\{50, 100, 200, 500\}$ across multiple random subsamples (five seeds for the boosting models, three for RandomForest) and compared each against a higher-budget reference ranking.

Table 8. Mean Spearman ρ between the ranking obtained at sample size n and the reference ranking.

n	XGBoost	LightGBM	RandomForest
50	0.991	0.989	0.983
100	0.995	0.991	0.989
200	0.997	0.996	0.992
500	0.998	0.998	0.994

Rankings are already stable at $n = 50$, and at the 200 instances used throughout the paper $\rho \geq 0.992$ for all three models. Across all 52 (size, seed, model) combinations, `sttl` was ranked first without exception.

Categorical encoding. Categorical features are label-encoded, which assigns arbitrary integer codes. To confirm that the tree models do not exploit an unintended ordinal structure in these codes, we reran the full pipeline with one-hot encoded `proto`, `service`, and `state`. For a like-for-like rank comparison, dummy importances are summed back onto their parent feature; otherwise `service`, spread over more than thirty sparse columns, would be mechanically penalised.

Table 9. Label encoding versus one-hot encoding: accuracy and pairwise Spearman agreement.

	Label encoding	One-hot
XGBoost accuracy	0.8735	0.8733
LightGBM accuracy	0.8812	0.8815
RandomForest accuracy	0.8828	0.8858
ρ XGBoost–LightGBM	0.861	0.861
ρ XGBoost–RandomForest	0.662	0.638
ρ LightGBM–RandomForest	0.647	0.741

Accuracy moves by at most 0.3 points, `sttl` remains rank 1 for all three models, and the boosting–boosting agreement is unchanged to three decimals. One difference is worth stating rather than glossing: the LightGBM–RandomForest pair rises by 0.094. This leaves the two-regime structure intact (boosting–boosting ≈ 0.86 versus RandomForest pairs ≈ 0.64–0.74) and affects no conclusion, but it is a real movement and we report it as such.

4.8. Per-Category Analysis

A dominant feature in an aggregate ranking may reflect a single high-frequency attack class rather than a dataset-wide property. We therefore recomputed SHAP separately on each attack category of the official test partition, capping each category at 200 instances so that rare and frequent categories are weighted comparably.

Table 10. Rank of `sttl` within each attack category of the official test partition.

Category	**LightGBM**	**RandomForest**	**XGBoost**
Normal	1	1	1
Exploits	1	1	1
Fuzzers	1	1	1
Reconnaissance	1	1	1
Shellcode	1	1	1
Worms	1	1	1
DoS	1	1	2
Generic	1	1	2
Analysis	1	1	3
Backdoor	1	1	3

The dominance is uniform. For LightGBM and RandomForest, `sttl` ranks first in all ten categories; for XGBoost it ranks first in six and within the top three in the remainder. The TTL cluster contributes between one and three members to the top-5 of every category.

The first row deserves emphasis: `sttl` ranks first for the *Normal* class as well. An attack signature would be expected to matter for the attack classes that exhibit it. A topology artifact that separates the two classes symmetrically is expected to matter everywhere, including for normal traffic, which is what we observe.

5. Discussion

5.1. What the Consensus Reveals and What It Conceals

The quantitative agreement between XGBoost and LightGBM ($\rho = 0.861$, 100% top-5 overlap) might initially appear reassuring: two independently designed algorithms, trained with different tree-growth strategies, converge on the same feature hierarchy. In ensemble design, such cross-model consensus is often interpreted as evidence that the shared features carry genuine predictive signal.

Our results complicate this interpretation. The feature on which all three models agree most strongly, `sttl` (source-to-destination TTL), is a network-layer value that reflects the operating system and routing configuration of the traffic source, not any intrinsic property of an attack. In the UNSW-NB15 lab environment, the attack traffic was generated by specific tools running on specific machines, producing characteristic TTL values that are statistically separable from the normal traffic generated by the same testbed. A model trained on this data can achieve high accuracy by learning this TTL signature, without having learned anything about what makes network behaviour malicious.

This is not a flaw specific to one model: all three classifiers, with different architectures and different inductive biases, converge on the same shortcut. The consensus, in this case, is a consensus on a shared artifact. An independent line of work supports this interpretation: Mohale and Obagbuwa [14], working on UNSW-NB15 with

a different set of classifiers (XGBoost, CatBoost, Decision Trees) and a different explanation portfolio (SHAP, LIME, ELI5), also surface `sttl` as the dominant indicator of malicious activity. Their finding (reached on the same dataset but through a different methodological route) reinforces our argument that the `sttl` signal is a property of UNSW-NB15 rather than of any particular model. This finding extends the argument of Corea et al. [10], who observe that most IDS classifiers rely on very few features, by showing that the features they rely on may not be semantically meaningful, and that quantitative agreement metrics are necessary to make this visible.

5.2. Boosting Models vs. RandomForest

The $\rho \approx 0.65$ agreement between boosting models and RandomForest is worth examining in detail. All three rank `sttl` first, but their secondary choices split: XGBoost and LightGBM move toward `sbytes` (source bytes) and `proto` (protocol), while RandomForest stays closer to TTL-family features, placing `dttl` (destination TTL) third and `rate` fourth. Boosting learns to combine the TTL signal with volume and protocol features; RandomForest leans more exclusively on TTL-based separation.

For ensemble IDS design this matters. XGBoost and LightGBM combined (a common pattern in stacking pipelines) bring little real diversity: they share the same dominant feature and will fail together on traffic where TTL values are not informative. RandomForest adds some diversity ($\rho \approx 0.65$ vs. $\rho = 0.86$), but not enough to cover the TTL blind spot. Genuine architectural diversity requires stepping outside the tree family.

5.3. Implications for IDS Evaluation and Deployment

These findings have three concrete implications for the IDS research community.

High accuracy does not validate features. All three models achieve 87–88% accuracy on the UNSW-NB15 test set while relying heavily on a TTL artifact. Accuracy measured on a held-out split from the same distribution cannot distinguish between a model that has learned generalizable attack patterns and one that has learned dataset-specific shortcuts. Cross-dataset transfer experiments, where a model trained on UNSW-NB15 is evaluated on NSL-KDD or CIC-IDS2017, provide a more demanding test of generalisation [12].

Cross-model SHAP agreement is a diagnostic tool. Our results suggest that Spearman rank correlation and top-k overlap between model pairs should become standard reporting metrics in IDS papers that use SHAP. If multiple models agree strongly on a small set of features, those features warrant manual inspection: are they semantically meaningful attack indicators, or statistical regularities of the benchmark?

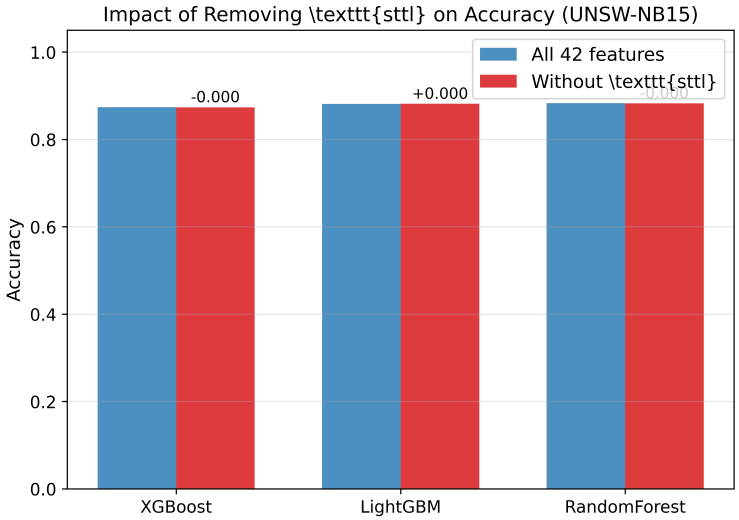


Figure 7. Accuracy with all 42 features vs. without `sttl`, on UNSW-NB15 (in-domain). Accuracy change is $\leq 0.04\%$ for all three models.

Dataset artifacts are model-agnostic. Because `sttl` dominates across three architecturally distinct classifiers, the artifact is not a property of any individual model but a property of the dataset. To verify this interpretation, we trained each model on all 42 features and again on the 41 remaining after removing `sttl`, then compared accuracy on the UNSW-NB15 test set. The accuracy change was negligible for all three models: -0.04% for XGBoost, $+0.03\%$ for LightGBM, and -0.01% for RandomForest (Figure 7). Removing the single most important SHAP feature changes nothing because the model immediately compensates via `ct_state_ttl` and `dttl`, two other TTL-derived variables present in UNSW-NB15. The shortcut is not encoded in `sttl` alone; it is distributed across a cluster of correlated TTL features, all of which reflect the same lab topology bias. Removing any one of them simply redistributes SHAP weight to the others.

5.4. On the 87–88% vs. 97% Accuracy Gap

The 87.35–88.28% accuracies we report on UNSW-NB15 sit below the 97%+ figures frequently quoted in the literature for the same dataset [8, 9]. This gap is not a deficiency of our models but a direct consequence of the evaluation protocol. Several published pipelines redraw a fresh random train/test split from the union of the official training and testing partitions, which mixes records originally separated by the dataset authors and inflates the test score due to the resulting distributional homogeneity [12]. By using the official split unchanged (175,341 train / 82,332 test), no record from the original test partition is ever seen during training, and the reported accuracies become directly comparable across studies that follow the same convention [10]. Reporting against the official partition is more conservative but, in our view, the only protocol that supports honest cross-paper comparison on UNSW-NB15.

5.5. The Effect of SMOTE on SHAP Explanations

Because SHAP values depend on the model's learned response surface, oversampling the training data with SMOTE [13] can in principle bias the feature attributions: synthetic minority samples occupy regions of feature space that were not represented in the original data, and any decision rule the model learns in those regions contributes to the SHAP value. Three considerations mitigate this concern in our setting. First, SMOTE is applied only to the training set; SHAP values are computed exclusively on the official, untouched test partition. The samples used to estimate $\Phi^{(m)}$ are therefore drawn from the natural data distribution, not the synthetic one. Second, the dominance of `sttl` is robust across architectures with very different sensitivities to local interpolation: gradient boosting (XGBoost, LightGBM, which split on quantile-binned thresholds) and bagging-based RandomForest (which averages many decorrelated trees) all rank `sttl` first by a large margin. A bias introduced by SMOTE in a smooth interpolation region would be unlikely to survive consistently across these inductive biases. Third, the UNSW-NB15 training set is only mildly imbalanced before SMOTE (56% attack, 44% normal), so the amount of synthetic minority data injected is small relative to the natural distribution. We therefore retain SMOTE for comparability with prior benchmarks [12] but interpret the SHAP rankings as reflecting model behaviour on the real test distribution.

5.6. The Physical Origin of the TTL Artifact

Attributing the consensus to a "lab topology artifact" is only convincing if the mechanism can be shown. The empirical distribution of `sttl` makes it explicit. Benign flows concentrate on the values 31 (60.4%), 254 (28.3%), and 62 (6.8%); attack flows concentrate on 254 (86.5%) and 62 (13.2%).

Conditioning in the other direction is sharper still. Of the 56,157 flows in the dataset with $\texttt{sttl} = 31$ (21.8% of all traffic), *not one* is an attack. For $\texttt{sttl} = 62$ the attack probability is 0.776, and for $\texttt{sttl} = 254$ it is 0.844. The first of these is not a statistical tendency but a deterministic rule embedded in the data.

The observed values cluster one to two hops below the initial TTL values 32, 64, and 255. The two classes were therefore emitted by hosts configured with different initial TTLs: benign traffic predominantly from machines initialised at 32, attack traffic from a source initialised at 255, a default typical of network appliances and traffic generators rather than of end-user operating systems. We state the distributions as measurements and the emission

mechanism as the inference they support; we do not assert a specific hardware configuration of the UNSW-NB15 testbed, which we could not verify against a primary source.

The practical consequence can be quantified without any model. A single-threshold rule, $\texttt{sttl} > 60 \Rightarrow$ attack, achieves 76.6% accuracy ($F_1 = 0.825$) on the official test partition, against 87.4–88.3% for the fully trained 42-feature ensembles. Most of the separability that these benchmarks reward is available from one byte of a network header that encodes which machine emitted the traffic.

5.7. Computational Cost

The cost of the diagnostic is dominated by the SHAP computation and varies by three orders of magnitude across model families. Under our protocol, TreeExplainer requires approximately 0.96 s per explained instance for the 500-tree, unbounded-depth RandomForest (roughly 6,300 leaves per tree), against milliseconds per instance for the gradient-boosting models. Cost is linear in the number of explained instances and in the number of trees, and grows with tree depth.

This is the constraint behind the 200-instance evaluation budget, and Section 4.7 is what justifies it: since rankings are already stable at $n = 50$, the expensive regime buys nothing. For boosting models the diagnostic is effectively free and can be run routinely; for deep forests it requires deliberate budgeting, and practitioners should subsample rather than explain a full test partition.

6. Conclusion

This paper quantified cross-model SHAP agreement between XGBoost, LightGBM, and RandomForest on UNSW-NB15, using Spearman rank correlation, Kendall τ, and top-k overlap to measure consensus across all 42 features. The results are clear on two counts.

First, the boosting models agree strongly with each other ($\rho = 0.861$, 100% top-5 overlap) and moderately with RandomForest ($\rho \approx 0.65$). The gradient boosting pair shares both a dominant feature and most secondary features; the boosting-RF pairs diverge on secondary features while converging on the top rank. Adding a multi-layer perceptron shows that this level of agreement is itself partly a property of the tree family: the MLP–boosting pairs fall to $\rho = 0.480$ and 0.536, the weakest in the study. Rank agreement across models of the same family should therefore not be read, on its own, as evidence about the data.

Second, and more importantly, $\texttt{sttl}$, the feature all three models agree on most, is a TTL artifact of the UNSW-NB15 lab topology. It has no semantic connection to attack behaviour; it reflects the specific machines used to generate attack traffic in that testbed. The mechanism is directly observable: of the 56,157 flows with $\texttt{sttl} = 31$, none is an attack, and a single-threshold rule on this one feature reaches 76.6% accuracy against 87.4–88.3% for the full 42-feature ensembles. The artifact survives every robustness check we applied: a change of model class, a change of attribution method, a change of categorical encoding, variation in SHAP sample size and seed, and disaggregation by attack category. When cross-model consensus points to a feature like this, the consensus is not evidence of genuine learning but evidence of a shared bias. Accuracy metrics cannot surface this problem; SHAP comparison can.

Scope of these claims. The empirical findings reported here concern UNSW-NB15. The identity of the consensus feature is a property of that benchmark's collection environment, and we make no claim that TTL artifacts are present in other intrusion-detection benchmarks; establishing whether the phenomenon recurs elsewhere requires independent semantic inspection of each dataset and is the subject of separate work. What we do propose as general is methodological rather than empirical: that agreement metrics be reported alongside accuracy, and that consensus features be inspected semantically before a benchmark result is trusted. We likewise make no claim that this diagnostic improves detection accuracy, robustness, or model selection: it is a screening step applied to benchmarks, not a method for building better detectors.

Limitations. SHAP values were computed on 200 stratified test samples, a budget justified in Section 4.7 but one that does not support per-instance analyses. All findings are specific to UNSW-NB15 and to binary classification. The non-tree model included here is a multi-layer perceptron; architectures whose inductive bias is tailored

to sequential or relational structure, such as recurrent, transformer, or graph-based models, remain untested. Hyperparameters were fixed to those of an established prior benchmark so that cross-model comparison would not be confounded by unequal tuning effort, and their influence on agreement was therefore not investigated.

Future work. Running the same comparison on NSL-KDD and CIC-IDS2017 would show whether TTL dominance is a peculiarity of UNSW-NB15 or a broader pattern in network traffic benchmarks. Removing `sttl` before training and measuring the effect on cross-dataset transfer would either confirm or refute the artifact hypothesis. Extending the present binary-classification setting to a multiclass formulation, in which each of the nine UNSW-NB15 attack categories is treated as a distinct label, would test whether the cross-model SHAP consensus on `sttl` persists across attack classes or splits into category-specific feature hierarchies; this multiclass setting is also closer to operational deployment. Finally, extending the analysis to recurrent and transformer architectures via KernelSHAP would test whether the same shortcut is exploited outside the tree family under inductive biases further removed from the ones examined here.

REFERENCES

1. N. Moustafa and J. Slay, *UNSW-NB15: A comprehensive data set for network intrusion detection systems*, in Proc. MilCIS, pp. 1–6, 2015.
2. M. Tavallaee, E. Bagheri, W. Lu, and A. A. Ghorbani, *A detailed analysis of the KDD CUP 99 data set*, in Proc. IEEE CISDA, pp. 1–6, 2009.
3. T. Chen and C. Guestrin, *XGBoost: A scalable tree boosting system*, in Proc. ACM KDD, pp. 785–794, 2016.
4. G. Ke, Q. Meng, T. Finley, T. Wang, W. Chen, W. Ma, Q. Ye, and T.-Y. Liu, *LightGBM: A highly efficient gradient boosting decision tree*, in Proc. NeurIPS, pp. 3149–3157, 2017.
5. L. Breiman, *Random forests*, Machine Learning, vol. 45, pp. 5–32, 2001.
6. S. M. Lundberg and S.-I. Lee, *A unified approach to interpreting model predictions*, in Proc. NeurIPS, pp. 4766–4777, 2017.
7. S. M. Lundberg, G. Erion, H. Chen, A. DeGrave, J. M. Prutkin, B. Nair, R. Katz, J. Himmelfarb, N. Bansal, and S.-I. Lee, *From local explanations to global understanding with explainable AI for trees*, Nature Machine Intelligence, vol. 2, pp. 56–67, 2020.
8. R. Younisse, A. Ahmad, and Q. Abu Al-Haija, *Explaining intrusion detection-based convolutional neural networks using Shapley additive explanations (SHAP)*, Big Data and Cognitive Computing, vol. 6, no. 4, p. 126, 2022.
9. S. Sivamohan and S. S. Sridhar, *An optimized model for network intrusion detection systems in industry 4.0 using XAI based Bi-LSTM framework*, Neural Computing and Applications, vol. 35, pp. 11459–11475, 2023.
10. P. M. Corea, Y. Liu, J. Wang, S. Niu, and H. Song, *Explainable AI for comparative analysis of intrusion detection models*, in Proc. IEEE MeditCom, 2024.
11. L. Grinsztajn, E. Oyallon, and G. Varoquaux, *Why do tree-based models still outperform deep learning on typical tabular data?*, in Proc. NeurIPS, vol. 35, 2022.
12. M. Berhili, O. Chaieb, and M. Benabdellah, *Intrusion detection systems in IoT based on machine learning: A state of the art*, Procedia Computer Science, 2024.
13. N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer, *SMOTE: Synthetic minority over-sampling technique*, Journal of Artificial Intelligence Research, vol. 16, pp. 321–357, 2002.
14. V. Z. Mohale and I. C. Obagbuwa, *Evaluating machine learning-based intrusion detection systems with explainable AI: enhancing transparency and interpretability*, Frontiers in Computer Science, vol. 7, art. 1520741, 2025. doi:10.3389/fcomp.2025.1520741.
15. V. Z. Mohale and I. C. Obagbuwa, *A systematic review on the integration of explainable artificial intelligence in intrusion detection systems to enhance transparency and interpretability in cybersecurity*, Frontiers in Artificial Intelligence, vol. 8, art. 1526221, 2025. doi:10.3389/frai.2025.1526221.
16. O. Subasi, J. Cree, J. Manzano, and E. Peterson, *A critical assessment of interpretable and explainable machine learning for intrusion detection*, arXiv preprint arXiv:2407.04009, 2024.
17. N. Aljarrah, H. H. Shehadeh, R. A. Obeidat, and M. Tawfik, *Cross-attention feature fusion for interpretable zero-day malware detection*, Statistics, Optimization and Information Computing, vol. 15, no. 3, pp. 2164–2178, 2026. doi:10.19139/soic-2310-5070-2900.
18. N. Azizi, A. Jamali, and N. Naja, *Comprehensive study: Machine learning and deep learning approaches in intrusion detection systems*, Statistics, Optimization and Information Computing, vol. 15, no. 2, pp. 1416–1432, 2026. doi:10.19139/soic-2310-5070-2782.
19. A. Rahman, M. S. I. Khan, M. Z. A. Eidmum, P. Shaha, B. Muiz, N. Hasan, T. Debnath, D. Kundu, J. T. Tamanna, M. Sayduzzaman, and M. Rahman, *Stacked ensemble method: An advanced machine learning approach for anomaly-based intrusion detection system*, Statistics, Optimization and Information Computing, vol. 14, no. 1, pp. 434–453, 2025. doi:10.19139/soic-2310-5070-2352.
20. R. Geirhos, J.-H. Jacobsen, C. Michaelis, R. Zemel, W. Brendel, M. Bethge, and F. A. Wichmann, *Shortcut learning in deep neural networks*, Nature Machine Intelligence, vol. 2, no. 11, pp. 665–673, 2020.