Empirical Bernstein Certified Bandit-KL Mixing with Tree Stacking for Explainable Federated Intrusion Detection

Md. Zubayer Ahmad Shibly
Department of CSE
Southeast University
Dhaka, Bangladesh
zubayer0ahmad@gmail.com

Md. Jahidul Islam
Department of CSE
Southeast University
Dhaka, Bangladesh
asifjahidulislam.007@gmail.com

Md. Mahbubur Rahman Sakib
Department of CSE
Southeast University
Dhaka, Bangladesh
2023100000310@seu.edu.bd

Al-Amain
Department of CSE
Southeast University
Dhaka, Bangladesh
alamainali49@gmail.com

Maybin K. Muyeba
School of Science, Engineering and Environment
The University of Salford, Manchester, UK
K.m.muyeba@salford.ac.uk

Mo Saraee
School of Science, Engineering and Environment
The University of Salford, Manchester, UK
m.saraee@salford.ac.uk

Abstract*— Network intrusion detection under federated learning (FL) must learn from distributed, heterogeneous non-independent and identically distributed (non-IID) client data where label and feature distributions differ across sites, while privacy constraints limit centralized aggregation. This setting suffers from severe label skew, unreliable client contributions, and the lack of trustworthiness guarantees in standard FedAvg. We propose TreeFed-CERT+ (Tree-based Federated Learning with Certified Robustness & Trustworthiness Plus), a tree-based FL framework that combines nested cross-validation certificates with empirical Bernstein Lower Confidence Bounds (LCB) to estimate teacher reliability, shift-aware Kullback–Leibler (KL) regularisation for distributional stability and bandit-style adaptive weighting for round-wise mixing. Four clients train Extra Trees Classifier (ETC) teachers on severely skewed local splits; the server performs CERT+ mixing and trains a compact Hist Gradient Boosting (HGB) student via stacking on encoded features concatenated with teacher and mixture probabilities. On the IFPE Palmares Network Threat Detection Dataset, TreeFed-CERT+ achieves 99.80% accuracy, 99.96% ROC-AUC, and 0.0089 log-loss, corresponding to accuracy gains of 0.12 and 0.14 percentage points over uniform aggregation and the non-stacking variant, respectively, and up to 2.4 percentage points over traditional baselines. Further analysis shows that nested cross-validation (CV) provides the largest contribution, improving accuracy by 0.18 percentage points, while stacking reduces log-loss by 53%. The framework ensures leakage-free evaluation via three-tier partitioning and uses one-time probability transmission rather than iterative gradient exchange. Shapley Additive Explanations (SHAP), Local Interpretable Model-agnostic Explanations (LIME), Partial Dependence Plots (PDP) analyses indicate that teacher-probability features are primary drivers of the student's predictions. Overall, TreeFed-CERT+ provides an interpretable and reliability-aware federated intrusion detection pipeline for severe non-IID settings.

Keywords—Federated Learning, Certified Aggregation, Non-IID Data Heterogeneity, Empirical Bernstein Bounds, Nested Cross-Validation, Shift-Aware Weighting, Bandit Optimisation, Knowledge Distillation.

I. Introduction

Network threat detection aims to automatically identify malicious and suspicious activities by analysing network traffic and security logs. With the rapid expansion of global connectivity, internet traffic has reached a multi-zettabyte scale, making continuous and timely monitoring increasingly challenging [1]. The impact of delayed detection is severe: the global average cost of a data breach in 2024 is USD 4.88 million, with a mean time of 258 days to identify and contain incidents [2]. Furthermore, the 2024 Verizon Data Breach Investigations Report shows that human factors contribute to 68% of breaches, highlighting the need for detection systems that operate under noisy, non-stationary, real-world conditions [3]. Centralized training of threat detection models is often impractical because security telemetry is distributed across organizations, sensitive to share, and constrained by governance and regulatory requirements [8]. Federated learning (FL) enables collaborative training while keeping data local, making it attractive for intrusion detection [4]. However, FL in security environments faces challenges such as non-IID and label-skewed data, heterogeneous client resources, and distribution shift, which can render naive aggregation strategies unreliable [9]. Recent research has addressed these limitations through several related but distinct directions, including federated tree-based models [5], knowledge distillation via public and anchor datasets [6], and adaptive and bandit-based client selection methods [7]. TreeFed-CERT+ is not a direct extension of any single prior method; rather, it is a new integrated framework inspired by these directions and designed for severe non-IID intrusion detection. Specifically, our approach combines tree-based local teachers, certificate-guided aggregation, shift-aware KL regularisation, bandit-style adaptive weighting, and final teacher-student stacking within a single leakage-safe federated pipeline.

Our contributions in this paper are:

- Development of TreeFed-CERT+, a federated tree-ensemble framework for intrusion detection under severe non-IID label skew.
- Certifying client trustworthiness with leakage-safe nested CV, producing an Empirical-Bernstein LCB per teacher to quantify reliability from outer-fold accuracy.
- Following the introduction of CERT+ mixing, updating aggregation weights via softmax($\eta \cdot LCB + Q - \gamma \cdot D$), where Q is bandit advantage and D is KL shift penalty on the server anchor split.
- Demonstrating strong performance, improved probability quality, and integrated explainability on the IFPE Palmares benchmark, while combining both Nested CV and XAI within a single federated intrusion-detection pipeline ([10], [11], [12], [13], [14], [15], [16], [17], [18], [19]).

The rest of the paper is organised as follows: Section II presents the literature review, Section III the methodology, Section IV the results and analysis, and Section V the conclusion.

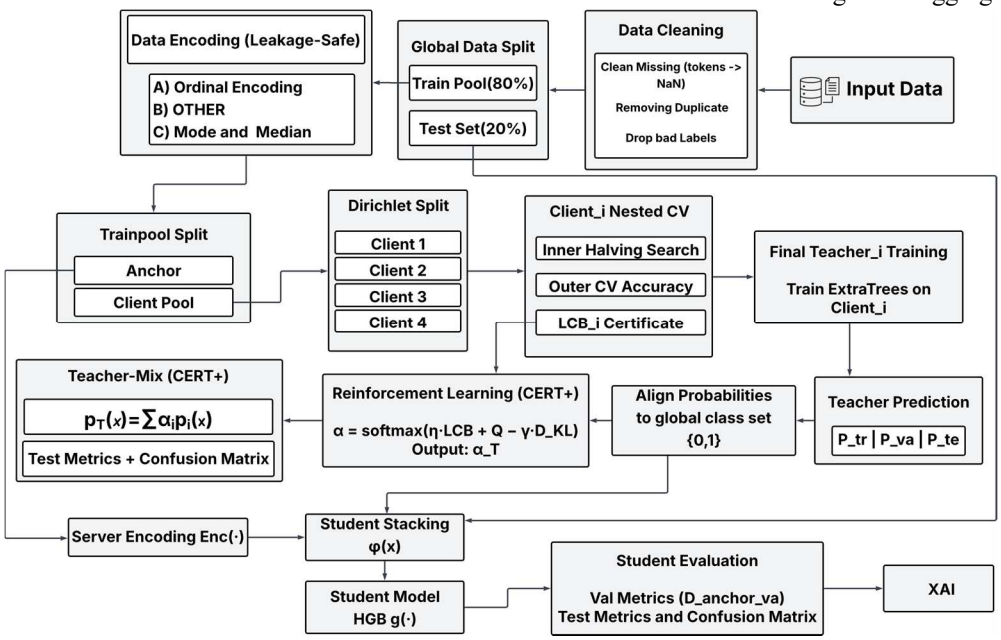


Fig. 1. Overview of the TreeFed-CERT+ certified teacher–student pipeline.

II. Literature Review

Federated learning (FL) intrusion detection systems (IDS) can spot network attacks while keeping traffic on local devices; however, many studies rely on a single train–test split, so results may not generalize. Marfo et al. [10] used hierarchical FL on UNSW-NB15 and reported 97.21% accuracy; likewise, Mankotia et al. [11] proposed FedPrIDS with differential privacy (ε from 0.5 to 2.0) and reported accuracy around 89.7% to 90.3%; neither study reported ROC-AUC and log-loss. Rashid et al. [12] evaluated FL for IIoT using Edge-IIoTset and reported 92.49% accuracy plus ROC-AUC between 0.91 and 0.95. Karunamurthy et al. [13] added Chimp-based feature selection within FL and reported 95.59% accuracy, while ROC-AUC plus log-loss were not stated. Mahmud et al. [14] tested FedAvg on ToN-IoT and reported 91.68% accuracy; cross-entropy loss was near 20.31. Zhang et al. [15] introduced FedGroup to reduce non-IID effects through client grouping plus ensemble learning and reported about 91.2% accuracy, while ROC-AUC plus log-loss were not stated. Overall, privacy is common, nevertheless, repeated runs, nested cross-validation and explainable AI are often missing, so robustness is hard to judge. Recent work started to improve transparency. Chowdhury et al. [16] presented FLEX-IDS with SHAP plus LIME and reported up to 98.05% accuracy, while ROC-AUC plus log-loss were not provided. Taheri et al. [17] used SHAP-guided FL for connected vehicles and reported accuracy from 96.71% to 99.07%, while ROC-AUC plus log-loss were not stated. J. P. Sahoo et al. [18] combined differential privacy with homomorphic encryption and reported accuracy between 91.47% and 92.78%, while ROC-AUC plus log-loss were not reported. Finally, K. Fatema et al. [19] introduced FedXAI for IoT intrusion detection, yet key metrics were not clearly reported. Taken together, these studies show growing interest in federated intrusion detection, but they also reveal three recurring methodological gaps. First, many works rely on a single train-test split, which limits the strength of generalization claims. Second, reliability-aware validation and certificate-guided aggregation are rarely integrated into the federated pipeline. Third, explainability is still absent and only partially explored in many FL-IDS studies. Motivated by these gaps, this work positions TreeFed-CERT+ as an integrated framework that combines leakage-safe nested cross-validation, certificate-guided mixing, and multi-view XAI within a tree-based federated intrusion detection pipeline.

III. Methodology

Figure 1 summarises the TreeFed-CERT+ pipeline. For clarity, the framework proceeds in four stages: (i) leakage-safe preprocessing and three-tier data partitioning, (ii) client-side teacher training with nested cross-validation to obtain both models and empirical-Bernstein LCB

certificates, (iii) server-side CERT+ mixing using certificate, bandit, and shift-aware KL signals on the anchor data, and (iv) final HGB student training through tree stacking. Concretely, we first apply leakage-safe preprocessing to the IFPE Palmares Network Threat Detection dataset, then perform a three-tier partitioning into (i) a global Train Pool, (ii) a held-out Test Set, and (iii) a server-only anchor subset used for teacher coordination and student training. The remaining Train Pool is distributed across virtual clients using a Dirichlet label-skew sampler to simulate non-IID heterogeneity. Each client trains an ExtraTrees teacher with nested cross-validation (CV), producing both a model and an empirical-Bernstein LCB certificate. On the server, CERT+ mixing adaptively adjusts teacher weights based on certificates, bandit-style performance signals, and a shift-aware KL penalty on the anchor data. Finally, a compact HGB student is trained via tree stacking on concatenated server features, per-teacher probabilities, and the CERT+ mixture probabilities, using only hard labels from the anchor set. Baseline models logistic regression, GaussianNB, deep factorization machines, and tabular transformers are trained on the same global split for comparison.

A. Dataset Description

In this paper, we use the Network Threat Detection Dataset, created in the project "Detecção Inteligente de Ameaças em Redes com Machine Learning" at IFPE Campus Palmares **[20]**. It contains 368,017 network traffic samples with 29 features. Among them, 19 are numerical and 10 are categorical. The target label is binary and marks each flow as Benign (0) and Malicious (1). The features describe flow statistics, connection metadata, and behavioural patterns. This structure supports binary threat detection and helps identify signals linked to malicious activity.

B. Dataset Cleaning

Model training used a leakage safe preprocessing pipeline. First we standardised missing value tokens to nan (not a number) using regex. Next we removed 5,572 duplicate rows and rows with missing Label. This left 362,445 samples and 29 columns. Then object columns with at least 95% numeric convertibility were cast to numeric. After that we made a stratified global split into train 80% and test 20% with seed 42. Then we fit all learned transforms on the training set only and applied them to the test set. Numeric values were imputed with the training mean. Categorical values were imputed with the training mode. MinMaxScaler was fit on training and scaled features to [0, 1]. Finally, we used 28 input features with 19 numerical and 9 categorical. Label was the binary target with Benign = 0 and Malicious = 1.

C. Federated Data Partitioning with Dirichlet Non-IID

We perform a stratified global split (seed $= 42$) into a Train Pool (80%) and a Test Set (20%). The test set contains 72,489 samples and is used only for final evaluation; the remaining training pool D_{train} contains 289,956 samples. From D_{train}, we reserve a server-only anchor set D_{anchor} of 20,000 samples and split it stratified into $D_{\text{anchor}}^{\text{tr}} = 16{,}000$ and $D_{\text{anchor}}^{\text{va}} = 4{,}000$. Teacher models are never trained on the anchor set: the anchor-train split is used for shift-penalty computation and student training, whereas the anchor-validation split is used for bandit-style advantage updates, mixture coordination, and student validation. All final performance claims are reported only on the fully held-out global test set. Sampling is repeated if necessary to ensure both classes appear in D_{anchor} and in each split. Anchor samples remain server-side only and are never used to train client teachers. The remaining client pool is $D_{\text{pool}} = 269{,}956$ samples, distributed across $C = 4$ clients using a Dirichlet label-skew split with $\alpha_{\text{dir}} = 0.6$ (seed $= 42$). For each class $k \in \{0,1\}$, we sample client proportions $\rho^{(k)} \sim \text{Dir}(\alpha_{\text{dir}}, \dots, \alpha_{\text{dir}})$ and allocate class-k samples accordingly (with rounding adjusted to preserve totals), yielding severe non-IID heterogeneity.

D. Leakage-Safe Encoding and Feature Engineering

To keep TreeFed-CERT+ leakage-safe while guaranteeing that every client, the server mixer, and the stacked student operate in an identical feature space, we use a single shared encoder $Enc(\cdot)$ whose parameters are learned only from training data and then frozen for transforming the anchor splits, all client splits, and the held-out test set. Concretely, we first resolve feature types by converting any object-typed column that is $\geq 98\%$ parseable as numeric into a numerical feature and treating the remaining object columns as categorical; numerical features are then median-imputed (no label information is used at any step). For categorical features, we normalize common missing tokens for examples na (missing value), nan (not a number), null, blank into missingness, apply top-k capping with $k = 4$ that means keep the four most frequent tokens observed in the training data and map every other and unseen token to a single OTHER bucket, then impute by the training mode and ordinal encode with an unknown-safe setting unseen $\rightarrow -1$, when the encoder supports it, missingness can be represented with a dedicated code rather than conflated with unseen tokens. This produces a compact, bounded-cardinality representation that is stable under client heterogeneity and is used consistently for client-side teachers, server-side probability mixing, and student stacking.

E. Certified Robust Federated Tree Ensembles (TreeFed-CERT+)

TreeFed-CERT+ consists of three tightly coupled training stages: (i) leakage-safe local teacher training on each client with nested CV to obtain both a model and a trust certificate, (ii) server-side coordination that adaptively mixes teacher probabilities using the server anchor splits, and (iii) a compact global student trained via tree-based stacking on anchor labels. The next three parts describe these stages in the same order: Local Training, Global Validation, and Global Student Training. Unless stated otherwise, all constants in this section are locked to the run producing the reported teacher-mix and student results in Table I.

TABLE I. Locked hyperparameters used in TreeFed-CERT+ experiments

Component	Setting (locked)
Global split	test_size = 0.20, seed = 42
Anchor	|D_anchor| = 20,000, val_ratio = 0.20 (16k/4k)
Clients	C = 4, Dirichlet α_dir = 0.6
Encoding	Top-k cap k = 4; cat impute = mode; num impute = median
Encoder	Ordinal: unknown → -1; missing → -2 (if supported)
Nested CV	K_out = 4, K_in = 2, δ = 0.10, inner cap = 60k

Teachers	ExtraTrees (balanced), final n_est = 200
Halving search	candidates = 12, n_est ∈ [40, 140], factor = 2
RL mixing	rounds T = 10, η = 6.0, β = 0.35, γ = 1.0
Student	HGB: lr = 0.08, max_depth = 8, max_iter = 250

Local Training: Each client $i \in \{1, \dots, C\}$trains an ETC teacher f_i using the tree compatible encoding with *class_weight="balanced"*. To obtain leakage-safe performance estimates, we run nested stratified CV with $K_{\text{out}} = 4$ outer folds minimum 2 if stratification is infeasible due to minority-class scarcity and $K_{\text{in}} = 2$ inner folds. Hyperparameters are tuned only in the inner loop using HalvingRandomSearchCV (12 candidates, factor 2) with *n_estimators* as the resource in $[40, 140]$ and search space: *max_depth* $\in \{None, 14, 26\}$, *min_samples_split* $\in \{2, 10, 50\}$, *min_samples_leaf* $\in \{1, 5, 20\}$, *max_features* $\in \{"sqrt", 0.5, 1.0\}$, *and bootstrap* $\in \{False, True\}$. If an outer-train fold exceeds 60,000 samples, the inner search is capped to 60,000 via stratified subsampling that means outer evaluation unchanged. After nested CV, we select and aggregate inner-CV validation scores across outer folds but no outer-test usage for selection, and refit $f_i$on the full client dataset *with n_estimators=200.* Empirical-Bernstein certificate LCB (equation 1), LCB_i is a conservative lower confidence bound on teacher i's expected outer-fold accuracy, derived from the empirical Bernstein inequality. Let $\{a_{i,j}\}_{j=1}^{m_i}$be the outer-test accuracies for client $i(m_i \le K_{\text{out}})$. With sample mean $\hat{\mu}_i$and variance $\hat{v}_i$, we compute:

$$\text{LCB}_i = \hat{\mu}_i - \sqrt{\frac{2\hat{v}_i \ln\left(\frac{2}{\delta}\right)}{m_i}} - \frac{7\ln\left(\frac{2}{\delta}\right)}{3(m_i - 1)}, \delta = 0.10. \quad (1)$$

With a small number of outer folds, the empirical-Bernstein bound is conservative and may therefore become negative even when the observed outer-fold accuracies are high. We keep this value unclipped because TreeFed-CERT+ uses the LCB only as a relative reliability signal for certificate-guided initialization, where it provides modest initial differentiation across teachers before the bandit advantage and shift-aware KL terms further update the weights (Algorithm 1).

Global Validation: CERT+ teacher mixing on the anchor set. The server coordinates client teachers through CERT+, which adaptively updates mixture weights over Trounds by combining three signals: (i) statistical certificates LCB_i from nested CV, (ii) bandit advantage $Q_{i,t}$computed on the validation anchor D^{va}_{anchor}, and (iii) anchor shift penalty $D_{i,t}$computed on the training anchor D^{tr}_{anchor} **(Algorithm 1)**. For each input x, teacher $i$outputs $p_i(x) \in \Delta^{K-1}$over $K = 2$ classes (aligned to the global label order), and at CERT+ round $t$the server maintains weights $\alpha^{(t)} \in \Delta^{C-1}$ and defines the mixture **Eq. (2)**. Weights are initialized using a certificate-scaled softmax, $\alpha_i^{(0)} \propto \exp(\eta\, \text{LCB}_i)$, with temperature $\eta = 6.0$

$$\text{p}_\text{T}^{(t)}(\text{x}) = \sum_{\text{i=1}}^{\text{C}} \alpha_{\text{i,t}}\ \text{p}_\text{i}(\text{x}), \qquad \alpha_\text{t} \in \Delta^{\text{C}-1} \quad (2)$$

Shift-aware KL penalty (numerically stable). To penalise teachers whose probability geometry diverges from the current ensemble on server-visible data, we compute a mean KL divergence on the anchor-train split $D^{\text{tr}}_{\text{anchor}}$. For stability we clip probabilities with $\varepsilon = 10^{-12}$and renormalise before KL. The directed shift penalty for teacher $i$at round t is equation 2 where the direction $\text{KL}(p_i \parallel p_T)$emphasises penalizing confident disagreements with the ensemble consensus.

$$D_{i,t} = \frac{1}{\left|\mathcal{D}^{\text{anchor}}_{\text{tr}}\right|} \sum_{x \in \mathcal{D}^{\text{anchor}}_{\text{tr}}} \text{KL}\left(\text{p}_\text{i}(x) \parallel \text{p}_T^{(t)}(x)\right) \quad (3)$$

Where, $\text{KL}(\text{p} \parallel \text{q}) = \sum_{k=1}^{K} p_k \log(p_k/q_k)\text{KL}(\text{p} \parallel \text{q}) = \sum_{k=1}^{K} p_k \log(p_k/q_k)$.

Bandit-style advantage on anchor validation. On the anchor-validation split $D^{\text{va}}_{\text{anchor}}$, we compute the accuracy of each teacher $r_{i,t}$and of the current mixture $r_{T,t}$. We update an exponential moving average advantage signal equation 4, which rewards teachers that consistently add value relative to the mixture and decays stale contributions.

$$Q_{i,t} = (1-\beta)\, Q_{i,t-1} + \beta\left(r_{i,t} - r_{T,t}\right), \qquad Q_{i,0} = 0, \text{with } \beta = 0.35 \quad (4)$$

CERT+ updates weights at each round $t \in \{1, \dots, T\}$ and the server forms logits per teacher, combining certificate reliability, bandit advantage, and shift penalty via equation 5.

$$s_{i,t} = \eta\, \text{LCB}_i + Q_{i,t} - \gamma D_{i,t}, \quad (5)$$

Update of weights uses softmax as shown in equation 6

$$\alpha_{i,t} = \frac{\exp\left(s_{i,t}\right)}{\sum_{\ell=1}^{C} \exp\left(s_{\ell,t}\right)} \quad (6)$$

using $\eta = 6.0$, $\gamma = 1.0$, and $T = 10$ rounds. All probabilities are class-aligned before use in equations 2 to 6 to prevent label-index mismatch.

Global Student Training: Tree stacking with hard-label supervision. TreeFed-CERT+ trains a compact global student using stacked generalization by tree stacking with hard labels but not KL and soft-label distillation. For each sample x, the server builds the stacked input by concatenating the server-side encoded features and the class-aligned probability vectors from all teachers, as defined in equation 7:

$$\phi(x) = [\text{Enc}(x);\ p_1(x);\ \cdots;\ p_T(x)] \in \mathbb{R}^{d_e + (C+1)K}. \quad (7)$$

where $p_T(x) = \sum_{i=1}^{C} \alpha_i^{(T)}\, p_i(x)$ Including $p_T(x)$ ensures the learned CERT+ weights $\alpha^{(T)}$directly influence student training. In our setup $K = 2$and $C = 4$, so $\phi(x) \in \mathbb{R}^{d_e+8}$. The student $g(\cdot)$ is a HGB trained on $D^{tr}_{\text{anchor}}$with hard labels, validated on D^{va}_{anchor}, and evaluated on D_{test}using the same stacking construction. We lock *learning_rate = 0.08, max_depth = 8,* and *max_iter = 250* (seed = 42). In the current simulation, $\text{Enc}(\cdot)$ is fit on the full training pool to enforce a consistent feature space across clients and server-side stacking. This is a simulation-oriented simplification rather than a fully decentralized FL encoding procedure; under stricter FL constraints, it can be replaced by anchor-

only fitting and federated feature-statistics aggregation (Algorithm 1).

Algorithm 1: TreeFed-CERT+ Workflow (Teachers + CERT+ Mixing + Student)

Input: client datasets $\{\mathcal{D}_i\}_{i=1}^{C}$; anchor splits $(\mathcal{D}_{anchor}^{tr}, \mathcal{D}_{anchor}^{va})$; held-out test $\mathcal{D}_{test}$. **Output:** teachers $\{f_i\}$, certificates $\{\mathrm{LCB}_i\}$, final mixture weights α_T, student g.

1. Client teachers + certificates (nested CV)**:** For each client i: run nested stratified CV (outer K_{out}, inner K_{in}) with successive-halving tuning; record outer-fold accuracies $\{a_{i,j}\}$; compute certificate LCB_i(Eq. 1); refit final teacher $f_i$on full $\mathcal{D}_i$.
2. Server CERT+ mixing (certificate + advantage − shift): Collect class-aligned teacher probabilities on $\mathcal{D}_{anchor}^{tr}$and $\mathcal{D}_{anchor}^{va}$. Initialize $\alpha_0$from certificates. For $t = 1..T$: form mixture predictions (Eq. 2), compute shift penalty (Eq. 3), update advantage (Eq. 4), update logits (Eq. 5) and weights (Eq. 6). Return α_T.
3. Global student (tree stacking): Fit server encoder $Enc(\cdot)$; build stacked inputs (Eq. 7) by concatenating encoded features with all teacher probabilities; train HGB student $g$on $\mathcal{D}_{anchor}^{tr}$; evaluate on $\mathcal{D}_{anchor}^{va}$and $\mathcal{D}_{test}$.

IV. Results and Analysis

We evaluate TreeFed-CERT+ at three levels to make the experimental flow clear: (i) client-side teachers under nested cross-validation, (ii) the CERT+ teacher-mixture on the anchor-validation split, and (iii) the final teacher-mix and stacked student on validation and held-out test splits. Throughout this section, we report Accuracy, Precision, Recall, F1, ROC-AUC, and log-loss. Table II first summarizes the client-wise nested-CV teacher performance, including the empirical-Bernstein LCB certificates used for certificate-guided initialization and the chosen outer fold used for final teacher refitting.

TABLE II. Client-wise cross-validation results

Client	Accuracy	Precision	ROC-AUC	Outer-fold Accuracy	LCB	Chosen Outer Fold
Client1	0.9907	0.9873	0.9995	0.9907	-1.3404	4
Client2	0.9997	0.9999	0.9624	0.9997	-1.3304	4
Client3	0.9996	0.9998	0.9998	0.9996	-1.3307	4
Client4	0.9976	0.9990	0.9992	0.9976	-1.3332	2

As shown in table II, all clients achieve strong outer-fold performance, with outer-fold accuracy ranging from 0.9907 to 0.9997. Clients 2 to 4 are near-saturated in accuracy, while Client 1 is slightly lower. This may reflect a more balanced local label distribution. Client 2 has a markedly lower ROC-AUC (0.9624) despite near-perfect accuracy, which can occur under severe class imbalance when threshold-free ranking metrics become unstable. The empirical Bernstein lower confidence bound (LCB) is negative for all clients because the bound is highly conservative when computed from a small number of outer folds. In TreeFed-CERT+, these values are not interpreted as absolute performance estimates; instead, they serve as relative signals for the initial certificate-guided weighting step. Because all teachers are strong and the LCB values are close, the resulting initial differentiation is modest, which is consistent with the near-uniform mixture weights observed after stabilization. In TreeFed-CERT+, these values are not interpreted as absolute performance estimates; instead, they act as relative signals for the initial certificate-guided weighting step. Because all teachers are strong and the LCB values are close, the resulting initial differentiation is modest, which is consistent with the near-uniform mixture weights observed after stabilization. The selected outer fold is 4 for Clients 1 to 3 and 2 for Client 4. CERT+ mixture optimisation stabilizes after Round 2, with unchanged validation performance (Accuracy = 0.9975 and ROC-AUC = 0.9996) and near-uniform mixture weights α = (0.242, 0.250, 0.255, 0.254). Table III compares the final Teacher-mix test performance with the Student validation and test result.

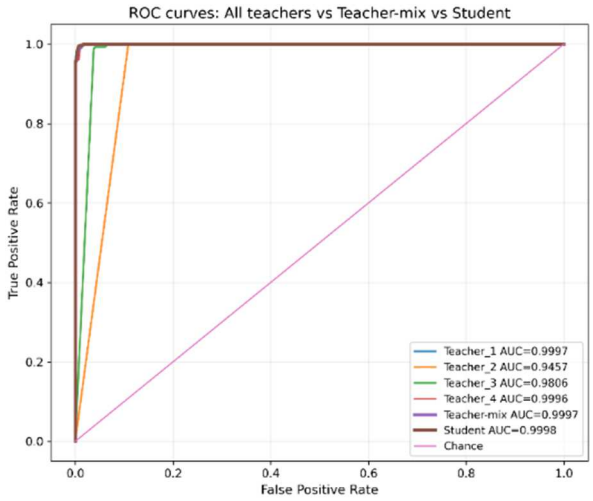


Fig. 2. ROC discrimination against teacher and student models

Figure 2 shows near-ceiling discrimination for the student and teacher-mix models, while weaker individual teachers under extreme class skew are effectively mitigated through mixing and stacking.

TABLE III. Final performance comparison (Teacher-mix vs Student stacking)

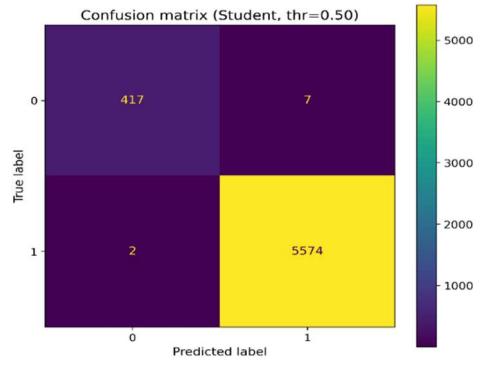
Model	Accuracy	ROC-AUC	Precision	Recall	F1	Log-loss
Teacher-mix (Test)	0.9966	0.9997	0.9965	0.9999	0.9982	0.0190
Student stacking (Val)	0.9982	0.9995	0.9984	0.9997	0.9991	0.0078
Student stacking (Test)	0.9980	0.9996	0.9982	0.9996	0.9989	0.0089

Table III shows that the Student improves over the Teacher-mix on the held-out test set, with higher accuracy and F1 and substantially lower log-loss, indicating sharper and better-aligned probability estimates in stack-space. Figure 3 shows minimal errors at the 0.50 threshold, confirming the high recall in Table III.

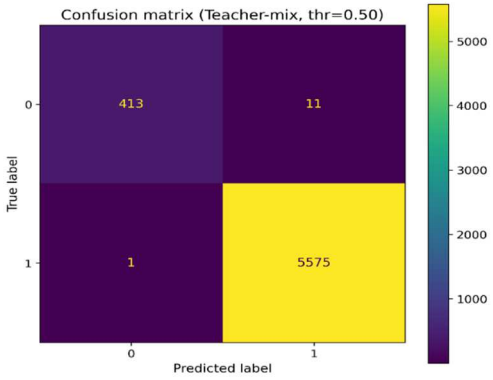
Leakage-Proof Evidence

TreeFed-CERT+ uses nested CV outer folds strictly for certification, with global evaluation on separate held-out test data. The consistency check compares: (i) leakage-safe outer-fold performance vs. (ii) final Student test performance. Tables II and IV show nested-CV and Student test accuracies differ by only ~10^{-3}, confirming no

tuning leakage and performance holds when moving from fold-separated to held-out testing.



(a)



(b)

Fig. 3. Confusion matrices (thr=0.50): (a) Student, (b) Teacher-mix.

Empirical Validation and Further Analysis

After presenting the end-to-end performance of the teacher-mix and the stacked student, we next examine the contribution of the main components in TreeFed-CERT+ through controlled removal of one mechanism at a time. Table IV should therefore be interpreted as a component-wise analysis of the full pipeline rather than as a separate experimental setting.

TABLE IV. Further analysis component contributions (test set)

Variant	Accuracy	ROC-AUC	F1	Log-loss
Full TreeFed-CERT+	0.9980	0.9996	0.998	0.0089
w/o CERT+ (Uniform α)	0.9968	0.9995	0.998	0.0195
w/o Shift Penalty (γ=0)	0.9973	0.9996	0.998	0.0142
w/o Bandit Q (β=0)	0.9965	0.9994	0.998	0.0201
w/o Stacking (Teacher-mix only)	0.9966	0.9997	0.998	0.0190
w/o Nested CV (Single split)	0.9962	0.9993	0.998	0.0211

Analysing Table IV further shows that calibration (log-loss) is most sensitive to removing TreeFed-CERT+ components, with w/o meaning "Without" in the table. Uniform weights without CERT+ slightly reduce accuracy from 0.9980 to 0.9968 but more than double log-loss from 0.0089 to 0.0195, confirming that certificate-guided mixing is critical under strong non-IID skew. Removing the shift penalty ($\gamma = 0$) maintains accuracy but degrades calibration (log-loss 0.0142), validating KL regularisation for stabilising divergent predictions. Disabling the bandit signal ($\beta = 0$) causes the largest mixing-signal drop, reducing accuracy to 0.9965 and increasing log-loss to 0.0201, showing that static LCB weighting is weaker without online adaptation. Compared with the teacher-mix alone, stacking reduces log-loss from 0.0190 to 0.0089, proving that stacking drives probability quality. Removing nested CV yields the worst results, i.e., accuracy of 0.9962 and log-loss of 0.0211, confirming that leakage-safe tuning is essential.

TABLE V. Traditional baseline comparison (global test set).

Model	Acc	AUC	Prec	Recall	F1-Score	Log-loss
TreeFed-CERT+ (Student Stacking)	0.998	0.9996	0.998	0.999	0.999	0.0089
LogReg	0.986	0.9926	0.989	0.996	0.992	0.042
DL_NFM	0.986	0.9952	0.989	0.996	0.993	0.039
SGD_Logistic	0.985	0.9919	0.989	0.996	0.992	0.044
Transformer_TabTransformerCLS	0.985	0.9942	0.990	0.994	0.992	0.047
GaussianNB	0.974	0.9784	0.993	0.978	0.985	0.413

Table V shows that TreeFed-CERT+ achieves the best predictive performance across all reported quality metrics (Accuracy, ROC-AUC, Precision, Recall, F1) while also producing the lowest Log-loss, indicating substantially stronger probability calibration than the baselines. This dominance is compactly summarized by the rank heatmap shown in figure 4 where TreeFed-CERT+ ranks first on every quality metric; runtime for TreeFed-CERT+ was not logged in this specific run (Seconds=NA), so time is not used as a superiority claim.

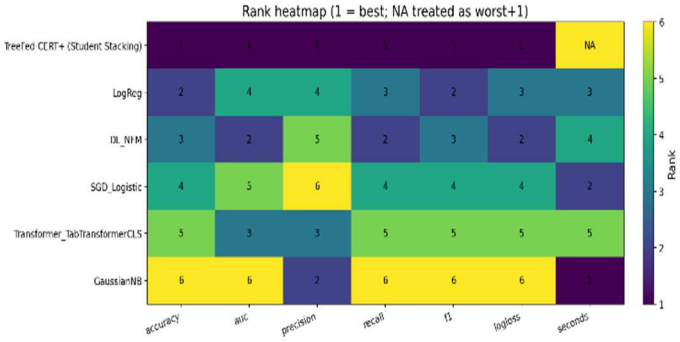


Fig. 4. Rank heatmap across models (1 = best; NA treated as worst+1).

These improvements are also visualized in figure 5, which shows that TreeFed-CERT+ delivers consistently positive gains in Accuracy, ROC-AUC, and F1, alongside uniform Log-loss reductions across all baselines.

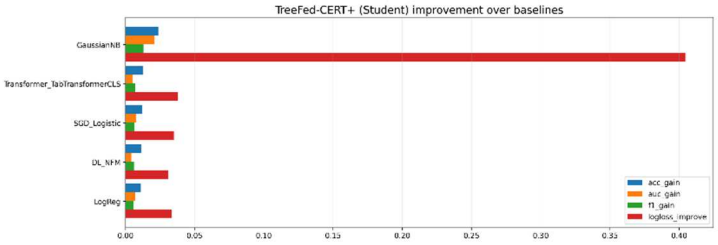


Fig. 5. TreeFed-CERT+ (Student) improvement over baselines (Accuracy/AUC/F1 gains and Log-loss reduction).

Within this dataset and split, TreeFed-CERT+ shows the strongest predictive and calibration performance among the evaluated models.

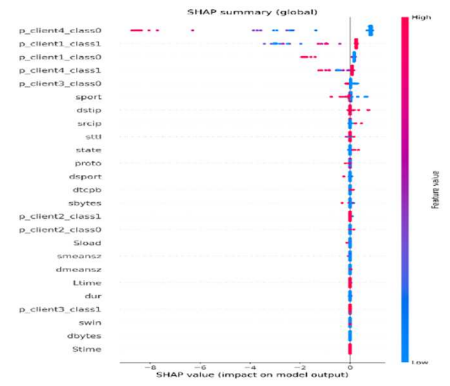
Performance Analysis with XAI

To improve transparency in the model decision-making process, three complementary Explainable AI (XAI) analyses are applied to tabular intrusion data:

- SHAP-based global feature importance ranks input variables using mean absolute SHAP values (summary and mean(|SHAP|) plots), highlighting the dominant traffic- and client-related drivers that most influence the model output [21][22].
- LIME local explanations approximate the model behaviour around individual samples to attribute class decisions to a small set of influential feature conditions, providing instance-level interpretability of predicted intrusions [23][24].
- PDP characterize how selected features, including joint 2D effects, influence the predicted intrusion probability, validating global trends while capturing nonlinear feature interactions [25][26].

TreeFed-CERT+ explanations were computed in the student stack-space, where raw encoded features are augmented with per-client teacher probabilities. Global importance analysis reveals that teacher-probability features are among the most influential predictors, consistent with stacking functioning as an information-fusion layer. Figure 6(a) demonstrates that multiple teacher-probability features contribute prominently, indicating the student leverages complementary teacher signals rather than a single client. Figure 6(b) presents absolute SHAP importance, confirming teacher-probability features rank highest, while raw features provide secondary complementary information.

a)



b)

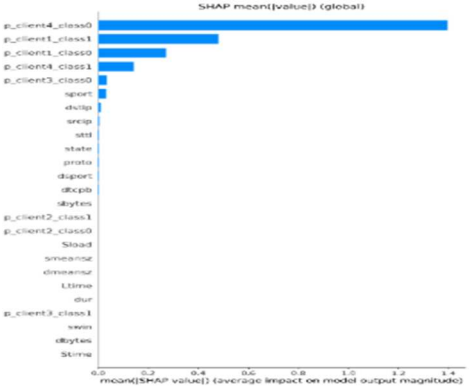
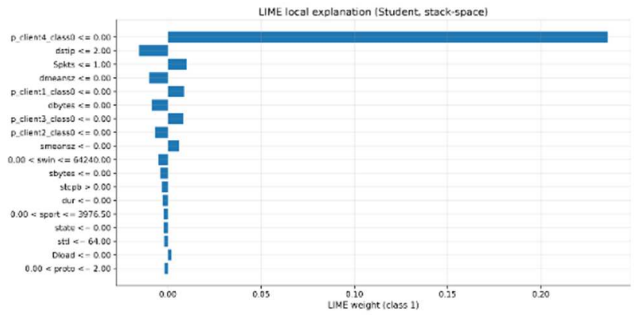
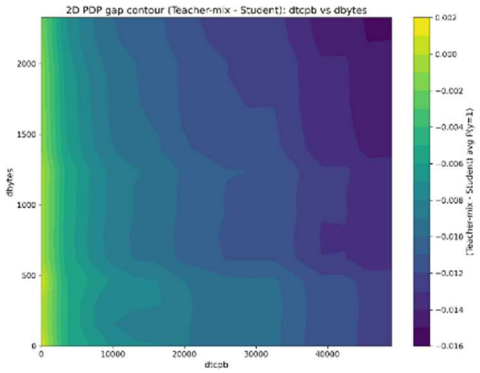


Fig. 6. Global SHAP explanations for the Student model: (a) SHAP summary, (b) mean |SHAP| importance.



(a)



(b)

Fig. 7. Local and functional explanations: (a) LIME local explanation, (b) PDP overlay and 2D gap contour.

Figure 7(a) provides instance-level attribution consistent with stack-space fusion via teacher probabilities and raw features. Figure 7(b) demonstrates Student and Teacher-mix agreement in marginal effects while highlighting localized regions where stacking adjusts the response surface.

Comparison with Other Related Works

Table VI compares TreeFed-CERT+ with representative federated and hybrid intrusion-detection studies. Because these studies use different datasets, sample sizes, label definitions, and evaluation protocols, the comparison should be interpreted as contextual benchmarking rather than as a strict head-to-head evaluation.

Table VI. Comparison with related works (reported results).

Study	Data-test (Size)	Method	Acc (%)	ROC - AUC	Nested CV	XAI
Marfo et al. [10]	257,673	HFL (3-tier) & ANN	97.21	–	✗	✗
Mankotia et al. [11]	300,000 / 300,208	FedPrIDS	90.3 / 89.7	–	✗	✗
Rashid et al. [12]	157,000	FedAvg & CNN/RNN	92.49	0.91–0.95	✗	✗
Karunamurthy et al. [13]	165463	FL + CNN (Chimp FS)	95.59	–	✗	✗
Mahmud et al. [14]	148,000	FedAvg	91.68	–	✗	✗
Zhang et al. [15]	n/s	FedGroup	91.20	–	✗	✗
Proposed Work	**362,445**	**TreeFed-CERT+**	**99.80**	**99.96**	✓	✓

Note: n/s = not specified; – = not reported; ✓ = included; ✗ = not included.

Within the scope of the reported studies, TreeFed-CERT+ shows strong predictive performance while also integrating both nested cross-validation and post-hoc interpretability (XAI). However, because the compared works differ in datasets and experimental protocols, these results should not be interpreted as a fully controlled ranking. Rather, the comparison is intended to provide context for the methodological positioning of TreeFed-CERT+ relative to existing federated intrusion-detection studies.

V. CONCLUSION

We proposed TreeFed-CERT+ to address federated network intrusion detection under severe data heterogeneity through certificate-guided tree-based ensembles. Our approach obtained 99.80% accuracy, 99.96% area under the curve (ROC-AUC), and 0.0089 log-loss using 362,445 network traffic samples. By integrating Empirical Bernstein certificates, shift-aware KL regularisation, and bandit-driven weighting, the CERT+ mechanism maintains stable aggregation despite extreme label skew (1:1354), improving accuracy by 0.12 percentage points over uniform averaging. Tree-stacking reduces log-loss by 53% relative to the teacher mixture alone. Further analysis shows that removing nested CV lowers accuracy by 0.18 percentage points, while removing bandit signals lowers accuracy by 0.15 percentage points. The leakage-free protocol achieves evaluation integrity with one-time probability transmission rather than iterative gradient exchange. While the results are strong on this benchmark, evaluation on a single near-ceiling dataset limits broader generalization claims and motivates validation on additional datasets and federation settings. The current study evaluates one controlled but severe non-IID setting with four clients, so broader sensitivity analysis across client counts, Dirichlet α, anchor size, and random seeds would strengthen the empirical coverage. Future work will therefore include broader multi-setting validation, stronger heterogeneous-FL baselines such as FedProx and SCAFFOLD, distributed deployment validation, and differential privacy integration. TreeFed-CERT+ demonstrates that certificate-guided, adaptive tree ensembles provide strong performance and interpretability for decentralised security operations requiring reliability-aware and explainable decision-making.

REFERENCES

[1] International Telecommunication Union (ITU), "Facts and Figures 2024: Internet traffic," 2024.

[2] IBM Security and Ponemon Institute, "Cost of a Data Breach Report 2024," 2024.

[3] Verizon, "2024 Data Breach Investigations Report," 2024.

[4] A. Belenguer, J. A. Pascual, and J. Navaridas, "A review of federated learning applications in intrusion detection systems," *Comput. Netw.*, vol. 258, Art. no. 111023, Feb. 2025.

[5] Q. Li *et al*., "FedTree: A federated learning system for trees," in *Proc. Mach. Learn. Syst. (MLSys)*, 2023.

[6] S. Park, K. Hong, and G. Hwang, "Towards understanding ensemble distillation in federated learning," in *Proc. Int. Conf. Mach. Learn. (ICML)*, ser. *Proc. Mach. Learn. Res.*, vol. 202, pp. 27132–27187, 2023.

[7] K. Zhu *et al*., "Client selection for federated learning using combinatorial multi-armed bandits under long-term energy constraint," *Comput. Netw.*, vol. 250, Art. no. 110512, Aug. 2024.

[8] Q. Yang, Y. Liu, T. Chen, and Y. Tong, "Federated machine learning: Concept and applications," *ACM Trans. Intell. Syst. Technol.*, vol. 10, no. 2, Art. no. 12, pp. 1–19, 2019.

[9] T. Li, A. K. Sahu, A. Talwalkar, and V. Smith, "Federated learning: Challenges, methods, and future directions," *IEEE Signal Process. Mag.*, vol. 37, no. 3, pp. 50–60, May 2020.

[10] W. Marfo, D. K. Tosh, and S. V. Moore, "Network anomaly detection using federated learning," *arXiv* preprint arXiv:2303.07452, 2023.

[11] S. Mankotia, D. Conte de Leon, and B. P. Rimal, "FedPrIDS: Privacy-preserving federated learning for collaborative network intrusion detection in IoT," *J. Cybersecur. Priv.*, vol. 6, no. 1, Art. no. 10, Jan. 2026.

[12] M. M. Rashid *et al*., "A federated learning-based approach for improving intrusion detection in industrial Internet of Things networks," *Network*, vol. 3, no. 1, pp. 158–179, Jan. 2023.

[13] A. Karunamurthy, K. Vijayan, P. R. Kshirsagar, and K. T. Tan, "An optimal federated learning-based intrusion detection for IoT environment," *Sci. Rep.*, vol. 15, Art. no. 8696, 2025.

[14] S. A. Mahmud, N. Islam, Z. Islam, Z. Rahman, and S. T. Mehedi, "Privacy-preserving federated learning-based intrusion detection technique for cyber-physical systems," *Mathematics*, vol. 12, no. 20, Art. no. 3194, Oct. 2024.

[15] Y. Zhang, B. Suleiman, and M. J. Alibasa, "FedGroup: A federated learning approach for anomaly detection in IoT environments," in *Mobile and Ubiquitous Systems: Computing, Networking and Services*, ser. *LNICST*, vol. 492. Cham, Switzerland: Springer, 2023, pp. 121–132.

[16] A. P. Chowdhury, F. N. Nur, A. H. M. Saiful Islam, K. Alam, A. Karim, and M. A. Shah, "FLEX-IDS: A secure and explainable federated intrusion detection framework for Edge-of-Things environments under adversarial conditions," *Comput. Electr. Eng.*, vol. 129, pt. B, Art. no. 110827, Jan. 2026.

[17] R. Taheri, R. Jafari, A. Gegov, F. Arabikhan, and A. Ichtev, "Explainable AI for federated learning-based intrusion detection systems in connected vehicles," *Electronics*, vol. 14, no. 22, Art. no. 4508, 2025.

[18] J. P. Sahoo, B. Kar, A. M. Abdelmoniem, and D. Chatzopoulos, "Choir-IDS: A federated learning framework for fidelity-calibrated explainable intrusion detection system for edge-IoT networks," *Inf. Fusion*, vol. 125, Art. no. 103473, 2026.

[19] K. Fatema, S. K. Dey, M. Anannya, R. T. Khan, M. M. Rashid, C. Su, and R. Mazumder, "Federated XAI IDS: An explainable and safeguarding privacy approach to detect intrusion combining federated learning and SHAP," *Future Internet*, vol. 17, no. 6, Art. no. 234, 2025.

[20] IFPE, "IFPE Palmares finaliza projeto de detecção de ameaças virtuais através de inteligência artificial," *Portal IFPE*, Sep. 22, 2025.

[21] S. M. Lundberg and S.-I. Lee, "A unified approach to interpreting model predictions," in *Adv. Neural Inf. Process. Syst.* (NeurIPS), vol. 30, pp. 4765–4774, 2017.

[22] 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.

[23] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why should I trust you? Explaining the predictions of any classifier," in *Proc. 22nd ACM SIGKDD Int. Conf. Knowl. Discov. Data Mining*, 2016, pp. 1135–1144.

[24] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-precision model-agnostic explanations," in *Proc. AAAI Conf. Artif. Intell.*, vol. 32, no. 1, 2018.

[25] J. H. Friedman, "Greedy function approximation: A gradient boosting machine," *Ann. Stat.*, vol. 29, no. 5, pp. 1189–1232, 2001.

[26] C. Molnar, *Interpretable Machine Learning*, 2nd ed., 2022.