Principled Federated Random Forests for Heterogeneous Data

Rémi Khellaf
PreMeDICaL team, Inria, Idesp
Inserm, Univ. de Montpellier
France
remi.khellaf@inria.fr

Erwan Scornet
Sorbonne Université
Université Paris Cité
CNRS, LPSM
F-75005 Paris

Aurélien Bellet
PreMeDICaL team, Inria, Idesp
Inserm, Univ. de Montpellier
France

Julie Josse
PreMeDICaL team, Inria, Idesp
Inserm, Univ. de Montpellier
France

Abstract

Random Forests (RF) are among the most powerful and widely used predictive models for centralized tabular data, yet few methods exist to adapt them to the federated learning setting. Unlike most federated learning approaches, the piecewise-constant nature of RF prevents exact gradient-based optimization. As a result, existing federated RF implementations rely on unprincipled heuristics: for instance, aggregating decision trees trained independently on clients fails to optimize the global impurity criterion, even under simple distribution shifts. We propose FedForest, a new federated RF algorithm for horizontally partitioned data that naturally accommodates diverse forms of client data heterogeneity, from covariate shift to more complex outcome shift mechanisms. We prove that our splitting procedure, based on aggregating carefully chosen client statistics, closely approximates the split selected by a centralized algorithm. Moreover, FedForest allows splits on client indicators, enabling a non-parametric form of personalization that is absent from prior federated random forest methods. Empirically, we demonstrate that the resulting federated forests closely match centralized performance across heterogeneous benchmarks while remaining communication-efficient.

1 Introduction

Random Forests (RFs) [3] are ensembles of CART decision trees [4] that recursively partition the feature space into regions, referred to as nodes, by selecting splits that maximize an impurity reduction criterion (e.g., variance or Gini). Predictions are then produced by aggregating the outcomes associated with the terminal nodes, commonly called leaves. Despite the deep learning revolution, RFs remain among the state-of-the-art and most widely used predictors for tabular data [6, 9, 13, 28], while also offering practical advantages: they naturally handle both continuous and categorical features, require minimal hyperparameter tuning, and are substantially more computationally efficient than deep learning models [9]. The strong empirical performance of RFs has been established in centralized settings, where data are assumed to be pooled together. However, this assumption is often unrealistic in many real-world applications, particularly in healthcare, where data are distributed across multiple institutions and cannot be centralized due to regulations, governance constraints, or competitive considerations [25, 1, 22, 32]. Federated learning (FL) addresses this challenge by enabling collaborative model training on decentralized data [14]. In this work, we focus on the

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

standard server–client architecture and on horizontal FL, where clients hold disjoint sets of records described by a common feature space.

Federating random forests presents challenges fundamentally different from those of parametric models, whose training can be federated by aggregating parameters or gradients across clients [see e.g., FedAvg 20]. In contrast, RFs are nonparametric and fully data-adaptive: each observation affects not only impurity values at each node but also the set of candidate split thresholds, which depend on the ordered feature values. Moreover, tree construction is inherently greedy and sequential, relying on hard, discontinuous splits; consequently, no gradients are available for aggregation across clients. To faithfully reproduce centralized behavior, impurity-based split decisions must be computed and updated using only aggregated client information, while respecting the sequential dependencies and without ever sharing raw data. This is made especially challenging by the heterogeneity of data across clients inherent to the federated setting: clients may differ in sample sizes, covariate distributions, or outcome mechanisms, and greedy split selection is highly sensitive to such differences. As a result, naive approaches—such as assembling locally trained forests or aggregating local split scores—can deviate substantially from centralized behavior.

Related work. A recent survey by [30] identifies three key challenges in decentralized training of tree-based models: identifying and computing sufficient split statistics, limiting communication costs, and handling client heterogeneity.

Decision tree-based federated learning has mostly focus on gradient-boosted decision trees (GBDTs), which federate training by aggregating histogram-based gradient statistics [16, 17] or by sequentially fitting local trees to the residuals of a global model [26]. GBDTs coordinate split selection using gradient information, whereas random forests provide no gradients and must rely on impurity-based, discrete split decisions. Moreover, boosted trees are inherently sequential in their construction: each tree depends on the residuals of all previous ones, and split selection typically requires full synchronization of clients at every node of every tree. In contrast, RFs are bagging-based, allowing trees to be constructed independently and in parallel—an important structural advantage in federated settings. Finally, federated GBDTs do not provide theoretical guarantees that the federated procedure reproduces the centralized greedy split objective.

Turning to federated RFs, early work focused on vertically partitioned data, using secure protocols for tree construction [19]. In the horizontal setting we consider, most approaches are *local-ensemble* methods: clients train trees or forests locally, and the server either aggregates predictions [18] or subsamples local trees to form a global ensemble [11, 5, 31, 10]. These heuristics are best suited to approximately i.i.d. data; they neither explicitly accommodate small clients (since each local tree is fit on a reduced per-client sample) nor address covariate shift or client-specific feature–outcome relationships, and are therefore highly sensitive to client heterogeneity.

To the best of our knowledge, no prior work federates the CART split-selection step *itself* in horizontal FL while maintaining a clear link to the centralized greedy objective and accounting for realistic client-level data heterogeneity. The closest approach is [15], which federates split selection by aggregating clients' local histograms at each node. However, their method is heuristic: it approximates centralized CART through binning, offers no theoretical guarantees on split fidelity, and is not designed to handle covariate or outcome shifts, nor outlier data points.

Although RFs enjoy a substantial theoretical foundation in the centralized setting [2, 27], analogous guarantees are currently lacking for their federated counterparts. To pave the way toward provably reliable federated forests, we study the fundamental question of whether a federated split-selection mechanism can recover, or closely approximate, the greedy split decisions made by centralized CART.

Our contributions. We propose a unified framework for federating CART decision trees and random forests that faithfully reproduces centralized training under realistic client heterogeneity without sharing individual data.

First, we introduce in Section 3.1 **a quantile-based candidate threshold generation scheme** and provide theoretical guarantees that it identifies the candidate closest to the split that would be selected by a centralized greedy CART. By estimating the pooled data distribution, defined as the mixture of the clients, this approach ensures that the threshold selection remains robust to client heterogeneity.

Next, we derive **an exact split-evaluation rule** (Section 3.2). We show that it reconstructs exactly the impurity gains that would result if all data were pooled, while relying solely on aggregated client statistics, thereby yielding decisions identical to the centralized CART algorithm. The key insight is that standard CART impurities admit additive sufficient statistics, which can be summed across clients to recover the centralized impurity—and hence the gain—exactly.

Finally, we design a method that allows trees to **split on the client indices** H, enabling the model to adjust on (X, H), instead of (X) only. To our knowledge, this is the first method to explicitly support such splits, enabling a nonparametric form of personalization.

We emphasize the effect of heterogeneous data on greedy split selection, showing that impurity gains depend not only on feature-outcome relationships but also on covariate distributions. Consequently, adding a client with a different marginal $P(X)$ can change split decisions even when $P(Y \mid X)$ is shared across clients. We also **propose *AvgImp*, a more efficient approach for homogeneous regimes** (approximately i.i.d.), which aggregates local impurity gains and is supported by a finite-sample error bound.

Finally, we leverage the federated CART split-decision rule introduced above to develop our theoretically grounded *FedForest* method. Using standard random-forest randomization (client-stratified bootstrapping and feature subsampling), *FedForest* grows trees in parallel from aggregated client statistics alone, **recovering centralized-level predictive performance** with communication-efficient training.

We validate these findings through benchmarks on synthetic and real datasets in Section 5. Together, our results provide a practical, theory-backed framework for federating nonparametric trees and random forests under heterogeneous data and lay the foundation for federated extensions of other random forest variants, including quantile [21], survival [12], and causal forests [29].

2 Problem Setting and Background

We consider a federated setting involving K clients collecting data described by the same set of features and addressing a supervised regression or classification task. Each client $k \in [K]$ holds a local dataset $\mathcal{D}_k = \{(H_i, X_i^{(k)}, Y_i^{(k)})\}_{i=1}^{n_k}$ consisting of n_k i.i.d. samples drawn from a d-dimensional distribution P_k, where the client indicator $H_i = k$ is constant within $\mathcal{D}_k$. The total sample size is $n = \sum_{k=1}^{K} n_k$.

Our goal is to learn a predictor $\hat{f} : \mathcal{X} \to \mathcal{Y}$ that performs well on the pooled population represented by the clients, i.e., the mixture distribution $P = \sum_{k=1}^{K} \rho_k P_k$ with weights $\rho_k = \mathbb{P}(H = k) \approx n_k/n$. More precisely, we aim to solve the following population optimization problem:

$$\min_{f \in \mathcal{F}} \ \mathcal{L}(f) = \sum_{k=1}^{K} \rho_k \, \mathbb{E}_{(X,Y)\sim P_k} \left[\ell\big(Y, f(X)\big) \right], \tag{1}$$

where ℓ is the squared loss for regression and the 0–1 loss for classification.

2.1 Centralized CART/RF Split Selection

The mixture-risk objective (1) is an ideal target. CART and Random Forests approximate it through greedy, node-wise optimization of an impurity criterion I (e.g., variance for regression; Gini or entropy for classification) on a finite data sample. In the centralized setting, all data are pooled as $\mathcal{D} = \bigcup_{k=1}^{K} \mathcal{D}_k$. At a node $\nu \subseteq \mathcal{X}$ of a tree, let $\mathcal{S}_\nu \subseteq \mathcal{D}$ be the samples reaching ν. CART selects a *split threshold* (j, t)–that is, a split variable $j \in [d]$ and a cutpoint $t \in \mathbb{R}$–in two steps: (i) *candidate generation*, where for each feature j, the values $\{x_{i,j}\}_{i \in \mathcal{S}_\nu}$ are sorted, and candidates t are defined as the midpoints between consecutive distinct values; and (ii) *split evaluation*, where each candidate (j, t) is scored by its empirical impurity reduction

$$\Delta I(j, t; \mathcal{S}_\nu) = I(\mathcal{S}_\nu) - \Big(\tfrac{n_L}{n_\nu} I(\mathcal{S}_L) + \tfrac{n_R}{n_\nu} I(\mathcal{S}_R) \Big), \tag{2}$$

with $\mathcal{S}_L = \{i \in \mathcal{S}_\nu : x_{i,j} \le t\}$, $\mathcal{S}_R = \{i \in \mathcal{S}_\nu : x_{i,j} > t\}$, $n_L = |\mathcal{S}_L|$, $n_R = |\mathcal{S}_R|$, and $n_\nu = n_L + n_R$. $I(\mathcal{S}_\nu)$ denotes the empirical impurity of the samples reaching node ν; for instance, in a regression setting, it stands for the empirical variance of the outcomes of samples falling into $\mathcal{S}_\nu$. CART then selects the threshold (j, t) that maximizes $\Delta I(j, t; \mathcal{S}_\nu)$.

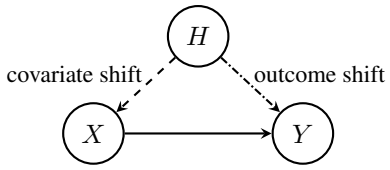


Figure 1: Graphical model of heterogeneity regimes. Dashed arrows indicate possible dependencies between the client variable H, the covariates and/or outcome mechanism, depending on the regime.

Random Forests build an ensemble of CART trees, adding randomness through bootstrap sampling—each tree is trained on a different sample of $\mathcal{D}$—and feature subsampling at each node (typically $\sqrt{d}$ for classification, $d/3$ for regression). Predictions are aggregated across trees by averaging in regression, or by majority vote in classification. In practice, individual trees are further regularized using standard stopping rules, such as maximum depth, minimum leaf size, or minimum impurity decrease.

In the federated setting, the pooled dataset is never accessible, and clients may only share aggregated summaries with the server. Existing federated tree methods largely restrict splits to X and lack a principled way to split on H. Our goal is thus to reproduce centralized split selection from aggregated client statistics while enabling client-aware splits under heterogeneity.

2.2 Heterogeneity Scenarios

We formalize the data heterogeneity regimes considered in this work and relate each of them to the behavior of a centralized CART-based model trained on pooled data. Figure 1 shows the associated graphical models and highlights situations where the client variable H carries predictive information for the outcome Y.

General model. In full generality, the combined data admit the decomposition

$$P(X, Y, H) = P(Y \mid X, H)\, P(X \mid H)\, P(H), \tag{3}$$

which allows both the covariate distribution and the outcome mechanism to vary across clients. This formulation captures realistic scenarios in which institutions differ both in their underlying populations and in the relationship between features and outcomes. The three regimes below correspond to restrictions of (3).

Assumption 2.1 (Homogeneous)**.** $(X, Y) \perp\!\!\!\perp H$.

In this scenario, all clients are drawn from the same population. Equivalently, $P(X \mid H) = P(X)$ and $P(Y \mid X, H) = P(Y \mid X)$. Pooling data only increases the effective sample size, while H carries no predictive information once X is observed. Consequently, splitting on H yields no (asymptotic) impurity reduction.

Assumption 2.2 (Covariate shift)**.** $P(Y \mid X, H) = P(Y \mid X)$, while $P(X \mid H)$ may depend on H.

In this regime, clients differ in their populations but share the same outcome mechanism; in other words, there is no "client effect". This implies that the conditional expectation of the outcome given features, $f(x) = \mathbb{E}[Y \mid X = x]$, is common to all clients, and therefore that H is not required for prediction once X is observed. However, the marginal feature distribution is a mixture $P(X) = \sum_{k=1}^{K} \rho_k P(X \mid H = k)$. As a result, clients may exhibit partially or even fully non-overlapping feature supports, so that the pooled dataset covers regions of the input space that are under-represented or entirely absent in individual local datasets.

Assumption 2.3 (Outcome shift)**.** $P(X \mid H) = P(X)$ while $P(Y|X, H)$ may depend on H.

In this setting, the conditional relationship of outcomes given the covariates may depend on the client. Conditioning on X alone is generally insufficient: the optimal prediction rule is $f(x, h) = \mathbb{E}[Y \mid X = x, H = h]$. Equivalently, H induces an effect on Y and must be included in the conditioning set. A centralized CART/RF trained on pooled data can capture this structure by treating H as an input feature and splitting on it whenever it reduces impurity.

3 Federated CART

In this section, we present our approach for reproducing CART's split selection in the federated setting. At any node ν, let $\mathcal{S}_{\nu,k}$ denote the subset of $n_{\nu,k}$ samples from client k that reach ν. The pooled node sample exists only as the virtual union $\mathcal{S}_\nu = \bigcup_{k=1}^K \mathcal{S}_{\nu,k}$. Our federated split selection rule has two key components: (i) constructing a finite set of candidate splits from distributed information (Section 3.1); and (ii) evaluating each candidate's impurity reduction using only client-to-server summaries (Section 3.2).

3.1 Candidate Split Generation

In centralized CART implementations [e.g., `scikit-learn` 24], the pooled values $\{x_{i,j}\}_{i\in\mathcal{S}_\nu}$ are sorted, and all midpoints between consecutive distinct values of each feature j are evaluated. While exact, this midpoint grid is impractical in the federated setting, as it would require communicating essentially all node-level feature values.

Naive alternatives are also unsatisfactory under client heterogeneity: fixed grids depend on feature scaling and can miss narrow high-gain regions, while random grids provide no guarantees and may miss low-density gaps between client supports under covariate shift. Instead, we propose *federated quantile sketching*: a data-adaptive, communication-efficient candidate-generation scheme that estimates the pooled (mixture) distribution at node ν from aggregated client summaries, and remains reliable under covariate shift, including disjoint client supports.

Federated quantile sketching. We construct a compact candidate set based on pooled quantiles. For each continuous feature j and node ν, client k computes B empirical quantiles $\mathbf{q}_{\nu,k}^{(j)} = (q_{\nu,k,1}^{(j)}, \ldots, q_{\nu,k,B}^{(j)})$ on its node data $\mathcal{S}_{\nu,k}$, and sends them to the server. The server converts $\mathbf{q}_{\nu,k}^{(j)}$ into a piecewise-linear approximation $\tilde{F}_{\nu,k}^{(j)}$ of the local empirical cumulative distribution functions (CDF) $F_{\nu,k}^{(j)}$, and then constructs a pooled CDF estimate via a mixture:

$$\tilde{F}_\nu^{(j)}(x) = \textstyle\sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} \tilde{F}_{\nu,k}^{(j)}(x), \tag{4}$$

with $n_\nu = \sum_{k=1}^K n_{\nu,k}$. Finally, candidate thresholds are selected as the interior quantiles of the reconstructed pooled CDF:

$$\mathcal{T}_\nu^{(j)} = \left\{ \left(\tilde{F}_\nu^{(j)}\right)^{-1}\left(\tfrac{b}{B}\right) : \; b = 1, \ldots, B-1 \right\}.$$

Importantly, we do not average local quantiles (which in general does not yield pooled quantiles); instead we aggregate CDF sketches and invert the pooled CDF.

Theorem 3.1 (Uniform rank error of the reconstructed CDF)**.** *Let $F_\nu^{(j)}(x) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} F_{\nu,k}^{(j)}(x)$ be the pooled empirical CDF for feature j at node ν, and let $\tilde{F}_\nu^{(j)}$ be obtained by linearly interpolating between the B reported quantile points per client and mixing as above. Then,*

$$\sup_x \left| \tilde{F}_\nu^{(j)}(x) - F_\nu^{(j)}(x) \right| \le \tfrac{1}{B}.$$

Corollary 3.2 (Approximation of centralized midpoint splits)**.** *Let $\mathcal{T}_{\text{cent}}$ denote the centralized midpoint candidates at node ν, and let $\mathcal{T}_{\text{fed}} = \bigcup_{j=1}^d \{(j,t) : t \in \mathcal{T}_\nu^{(j)}\}$ be the quantile-based grid. For any $(j, t_{\text{cent}}) \in \mathcal{T}_{\text{cent}}$, there exists $(j, t_{\text{fed}}) \in T_{\text{fed}}$ such that the induced left-child assignments disagree on at most a $\frac{3}{2B}$ fraction of pooled samples:*

$$\tfrac{1}{n_\nu} \textstyle\sum_{i\in\mathcal{S}_\nu} \mathbb{1}\left(\mathbb{1}(x_{i,j} \le t_{\text{cent}}) \ne \mathbb{1}(x_{i,j} \le t_{\text{fed}})\right) \le \frac{3}{2B}.$$

Theorem 3.1 and Corollary 3.2 (proved in Appendix C.1) show that B directly controls the rank resolution of the candidate grid: increasing B makes the federated quantile thresholds closer to the centralized midpoint thresholds in terms of induced child assignments. For example, setting $B = 100$ percentiles implies at most 1.5% disagreement in child assignments while requiring only $B - 1$ candidates per feature, independent of n_ν. These results allow precise control of the trade-off between communication cost and candidate-threshold fidelity to the centralized approach.

Our quantile-based candidates offer several advantages to the fixed-width histograms proposed by [15] for three reasons: (i) they adapt to local data density, ensuring critical split points are preserved

as well as possible for a given budget B; (ii) they ensure an efficient use of bins by enforcing a minimum sample fraction of $1/B$ within each bin; and (iii) they are more robust against outliers (see experiment in Section F.3). This latter point is increasingly important in settings with high client diversity, which naturally stretches the global covariate space.

Top-L feature shortlisting for homogeneous data. Under Assumption 2.1, local impurity gains concentrate around the pooled gain, so local rankings tend to agree with the global ranking. This enables a communication-efficient preselection step: each client k computes for each feature j its best local gain $M_{\nu,k}(j) = \max_t \ \Delta I(j, t; \mathcal{S}_{\nu,k})$ and sends only its top-L features (with associated gains). The server merges these reports (e.g., taking the union of the local top-L sets) to form a reduced feature set $\mathcal{L}_\nu$ on which it performs quantile sketching and split evaluation. A supporting guarantee for retaining the pooled-best feature under Assumption 2.1 is given in Appendix C.5.

Discrete features. For small-cardinality categorical variables, clients can send exact category counts and the server evaluates all admissible splits with negligible cost. For high-cardinality categories, we use the standard ordering trick known as Fisher grouping [7]: we order categories by their within-node outcome mean (regression) or class proportion (binary classification), treat the ordered categories as an ordinal feature, and scan contiguous splits.

3.2 Federated Impurity Reduction

Given the candidate set $\mathcal{T}_{\text{fed}}$ obtained in Section 3.1, the remaining task is to evaluate the impurity reduction (2) for each candidate split (j, t) without exchanging raw data. We first describe *AvgImp*, a natural evaluation rule for the homogeneous setting, and then introduce our main approach for the general heterogeneous setting.

Averaging local impurity gains under client homogeneity (*AvgImp*). Under Assumption 2.1, local split objectives are expected to align with the centralized CART objective. This motivates a simple federated evaluation rule based on averaging local impurity reductions. For each candidate split (j, t), client k computes its local gain $\Delta I(j, t; \mathcal{S}_{\nu,k})$, and the server forms the weighted average

$$\widehat{\Delta I}_{\text{AvgImp}}(j, t; \mathcal{S}_\nu) := \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \Delta I(j, t; \mathcal{S}_{\nu,k}).$$

The split is then selected by maximizing $\widehat{\Delta I}_{\text{AvgImp}}$ over the $\mathcal{T}_{\text{fed}}$. In practice, this can be combined with local Top-L screening: each client reports only its best candidates according to local gains, and the server evaluates the averaged criterion on the resulting shortlist.

Theorem 3.3 (*AvgImp* approximation error on homogeneous data)**.** *Under Assumption 2.1 and $\mathbb{E}[Y^2] < \infty$, we have:*

$$\left| \widehat{\Delta I}_{\text{AvgImp}}(j, t; \mathcal{S}_\nu) - \Delta I(j, t; \mathcal{S}_\nu) \right| = O_p\Big(\frac{K}{n_\nu}\Big).$$

While this approximation bound becomes uninformative in highly fragmented regimes where K is comparable to n, it is meaningful in typical cross-silo federated learning scenarios, where $n \gg K$ (i.e., a small number of clients each with large datasets, such as hospitals with patient records).

However, the guarantee above relies on Assumption 2.1. In heterogeneous settings, we show that averaging local impurities does not generally recover the pooled impurity, as described in the following decomposition.

Theorem 3.4 (General impurity decomposition)**.** *The impurity reduction for a given candidate decomposes as*

$$\Delta I(j, t; \mathcal{S}_\nu) = \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \Delta I(j, t; \mathcal{S}_{\nu,k}) + \Delta \mathcal{E}(j, t; \nu), \tag{5}$$

where $\Delta \mathcal{E}(j, t; \nu)$ is a split-dependent heterogeneity correction, given in Section C.3. It corresponds to the change in the Jensen-gap term $\mathcal{E}(\nu)$ caused by splitting the within-node mixture of client distributions. For standard strictly concave impurity functions (e.g., Gini index, entropy, and variance), we have $\mathcal{E}(\nu) \geq 0$, with $\mathcal{E}(\nu) = 0$ if and only if

$$\forall (k, k'), \quad \hat{\mathbb{E}}[Y \mid \mathcal{S}_\nu, H = k] = \hat{\mathbb{E}}[Y \mid \mathcal{S}_\nu, H = k'],$$

that is, the within-node empirical outcome means, or class probabilities, coincide across all clients.

Thus, when clients hold heterogeneous distributions, split decisions based on *AvgImp* will deviate from those produced by a centralized CART procedure. This discrepancy motivates an alternative evaluation rule that remains exact even under heterogeneous data.

Exact impurity evaluation via additive summaries. Our main evaluation rule relies on the novel observation that standard CART impurities admit additive sufficient statistics, as shown in Theorem 3.5. These statistics can be aggregated across clients to recover the centralized impurity reduction exactly for any candidate split.

Theorem 3.5 (Exact centralized evaluation via additive summaries)**.** *For standard CART criteria, there exists a deterministic map Ψ and per-client summaries $\mathbf{s}_{\nu,k}$ such that the pooled impurity reduction can be written as*

$$\Delta I(j,t;\mathcal{S}_\nu) = \Psi(\mathbf{s}_\nu) - \Big(\tfrac{n_L}{n_\nu}\Psi(\mathbf{s}_L) + \tfrac{n_R}{n_\nu}\Psi(\mathbf{s}_R)\Big), \tag{6}$$

with aggregated statistics $\mathbf{s}_m = \sum_{k=1}^{K} \mathbf{s}_{m,k}$ for $m \in \{\nu, L, R\}$.

In practice, for each candidate (j,t), client k sends only its left-child summary $\mathbf{s}_{L,k}$. The server then aggregates these summaries into $\mathbf{s}_L$, recovers $\mathbf{s}_R = \mathbf{s}_\nu - \mathbf{s}_L$, and evaluates (6). This procedure exactly replicates centralized split evaluation, irrespective of covariate or outcome shift.

Examples. For C-class classification, $\mathbf{s}_\nu = (n_{\nu,1}, \ldots, n_{\nu,C})$ and $\Psi_{\mathrm{Gini}}(\mathbf{s}_\nu) = 1 - \sum_{c=1}^{C}(n_{\nu,c}/n_\nu)^2$, so each client sends C scalars per candidate. For regression, $\mathbf{s}_\nu = (n_\nu, S_\nu, Q_\nu)$, with $S_\nu = \sum y_i$ and $Q_\nu = \sum y_i^2$, and $\Psi_{\mathrm{Var}}(\mathbf{s}_\nu) = Q_\nu/n_\nu - (S_\nu/n_\nu)^2$, so each client sends three scalars per candidate.

In summary, *AvgImp* provides a natural evaluation rule under homogeneous client distributions, but its guarantees do not extend to violations of Assumption 2.1. The additive-summary rule in Theorem 3.5 is therefore our primary evaluation rule when exact centralized split selection is required.

Outcome shift: splitting on H with no extra communication. When Assumption 2.3 holds (which can be tested in a federated manner, see Section F.4), the target rule is $f(x,h) = \mathbb{E}[Y \mid X = x, H = h]$, and accurately fitting this rule may require trees with splits on H. Let $\mathcal{K}_\nu = \{k : n_{\nu,k} > 0\}$ be the set of active clients at node ν. Once the server has collected the per-client node summaries $\{\mathbf{s}_{\nu,k}\}_{k\in\mathcal{K}_\nu}$—already needed to evaluate feature-based splits—it can also evaluate splits on H *at no additional communication cost*, i.e., without requesting any extra messages from clients. Indeed, any partition $\mathcal{K}_\nu = \mathcal{K}_L \cup \mathcal{K}_R$ induces $\mathbf{s}_{\nu,L} = \sum_{k\in\mathcal{K}_L} \mathbf{s}_{\nu,k}$ and $\mathbf{s}_{\nu,R} = \sum_{k\in\mathcal{K}_R} \mathbf{s}_{\nu,k}$, and the corresponding gain follows directly from (6) using the same map Ψ.

Although there are $2^{|\mathcal{K}_\nu|-1}$ partitions, the optimal split can be found efficiently: for regression and binary classification, it suffices to sort sites by their within-node means $\bar{y}_{\nu,k}$ and scan contiguous splits [7], for a cost $O(|\mathcal{K}_\nu| \log |\mathcal{K}_\nu|)$. When H has many categories, treating H-splits as Fisher groupings is essential; otherwise trees waste depth constructing arbitrary partitions of client IDs.

4 Federated Forests

We now show how to lift the federated CART primitives from Section 3—candidate generation and summary-based split evaluation—into a scalable Random Forest training protocol, which we call *FedForest*. The protocol, described in Algorithm 1, rests on three key components.

Client-stratified bootstrap. For each tree, client k draws (with replacement) a bootstrap sample of size n_k from its local data $\mathcal{D}_k$, and the tree is trained on the virtual union of these resamples across clients (Alg. 1, Line 4). This preserves the pooled mixture weights $\rho_k = n_k/n$ in every tree. In contrast, a naive pooled bootstrap can under-represent—or entirely omit—small clients.

Feature subsampling. As in standard Random Forests, we introduce tree-level randomness via feature subsampling to promote diversity across trees. For each active node, the server samples a subset of features $\mathcal{J}_\nu$ to evaluate (Alg. 1, Line 6). In the federated setting, this also reduces communication, as fewer feature–threshold pairs need to be evaluated and transmitted by each client. It can further be combined with Top-L shortlisting to achieve additional savings.

Level-wise batched training. Tree growth is sequential *within* each tree but independent *across* trees. We leverage this by training all trees level-wise (Alg. 1, Lines 5–18): at each depth, the server batches active nodes across all trees into a single request. Clients answer for the whole batch in parallel (Alg. 1,

Algorithm 1 FEDFOREST

1: **Input:** #trees T, depth M, client subsampling rate r, sketch size B.
2: **Output:** Forest predictor $\hat{f}$.
3: **Init:** Server initializes root nodes $\mathcal{A} = \{(\tau, \text{root})\}_{\tau=1}^{T}$.
4: For each τ, server selects client subsets $\mathcal{K}_\tau$ (size $\lceil rK \rceil$); Clients draw bootstrap samples.
5: **for** depth level $\ell = 0$ **to** $M - 1$ **do**
6: **Server:** Broadcasts previous split decisions and selects features $\mathcal{J}_\nu$ for active nodes $\nu \in \mathcal{A}$.
7: **Clients (Parallel):** For all τ and $k \in \mathcal{K}_\tau$:
8: Update local node membership for active nodes ν.
9: Compute sketches $\mathbf{q}_{\nu,k}^{(j)}$ for continuous $X \in \mathcal{J}_\nu$ and local node summaries $\mathbf{s}_{\nu,k}$ (for H).
10: Send sketches and summaries to Server.
11: **Server (Candidate Generation):**
12: **For X:** Aggregate sketches $\rightarrow$ approximate global CDF via Eq. 4 $\rightarrow$ candidates $\mathcal{T}_\nu^{(X)}$.
13: **For H:** Compute site means from $\mathbf{s}_{\nu,k}$; sort site indices by mean; generate splits $\mathcal{T}_\nu^{(H)}$.
14: Broadcast candidates $\mathcal{T}_\nu = \mathcal{T}_\nu^{(X)} \cup \mathcal{T}_\nu^{(H)}$.
15: **Clients (Parallel):** For all τ and $k \in \mathcal{K}_\tau$:
16: Compute **left-child summaries** $\mathbf{s}_{L,k}(j,t)$ for all candidates $(j,t) \in \mathcal{T}_\nu$ and send to Server.
17: **Server:** Aggregates $\mathbf{s}_L = \sum_k \mathbf{s}_{L,k}$; evaluates impurity gain via Equation (6);
18: Selects best split (j^*, t^*) and updates $\mathcal{A}$.
19: **Finalize:** Server computes leaf values from final node summaries.

Lines 7–11 and 15–17), and the server updates all nodes independently. As a result, training T trees of depth M requires only $O(M)$ synchronized rounds, rather than scaling with the total number of nodes.

Client subsampling (Optional). For a given tree, the server can randomly sample a client subset $\mathcal{K}' \subseteq [K]$ (Alg. 1, Line 4) to reduce communication. We recommend this mainly as an acceleration knob under approximately homogeneous clients. Under covariate shift, it changes the mixture distribution and may alter split decisions; under outcome shift, it can omit informative values of H.

Communication and computation costs. We summarize and compare the communication and complexities of different methods in Table 2, and derive the per-node scalar costs in Section A. It highlights that histograms broadcast only the selected split, reducing the communication cost by a factor of dB compared to quantile candidates. However, in high-dimensional settings, the top-L procedure used in *AvgImp* requires significantly less communication than histograms.

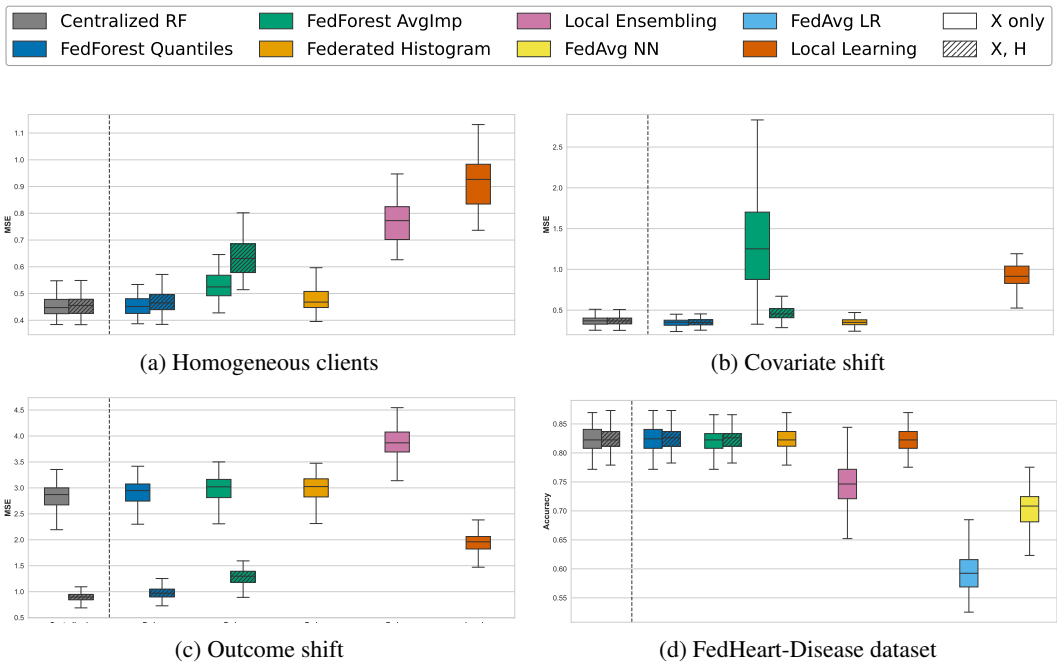
5 Numerical Results

Experimental setting. We consider a centralized `scikit-learn` Random Forest trained on pooled data as a reference, as well as two federated baselines reimplemented due to the lack of public code: Federated Histogram [15], adapted to regression via node-wise sufficient statistics for squared loss, and one-shot Local Ensembling [11], where clients train local forests and the server aggregates predictions. We also include Local Learning, where each client trains a forest on its own dataset $\mathcal{D}_k$. We compare these with two variants of our approach: FedForest-Quantiles, which combines quantile sketches with exact split evaluation via additive sufficient statistics, and FedForest-AvgImp/Top-L, which uses Top-L feature shortlisting followed by impurity evaluation based on averaged local gains.

All methods use $T = 50$ trees, maximum depth 8, minimum leaf size 5, feature subsampling ($d/3$ for regression, $\sqrt{d}$ for classification), and client-stratified bootstrapping. FedForest and Federated Histogram use $B = 32$ candidates per feature, while AvgImp/Top-L uses $L = 3$ screened features. For methods that split on both X and H, H is always included as a candidate (i.e., exempt from feature subsampling), whereas baselines without client splits use only X. Since standard regression forests do not natively handle unordered categorical splits on H, centralized RF uses Fisher encoding at the root node, while FedForest recomputes it at each node from shared summaries.

Synthetic data. We simulate a horizontal multicenter setting with $K = 10$ clients. Client k observes $n_k = 200$ i.i.d. samples $(H_i = k, X_i^{(k)}, Y_i^{(k)})$, where $X_i^{(k)} \sim \mathcal{N}(\mu_k, \alpha_k I_d)$ and $\mu_k = [(-1)^k\gamma, \ldots, (-1)^k\gamma]$. Here, $\gamma \geq 0$ and $\alpha_k > 0$ control covariate heterogeneity: smaller γ and larger

α_k induce stronger overlap across clients. Outcomes are generated as $Y_i^{(k)} = f(X_i^{(k)}) + \delta_k + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, .5)$ and $\delta_k = (-1)^k \delta$. Thus, δ controls outcome heterogeneity, while f is a tree-based regression function (see Section F). We set $d = 20$ and study three settings: (i) **homogeneous** ($\gamma = \delta = 0$, $\alpha_k = 1$), (ii) **covariate shift** ($\gamma = 3$, $\alpha_k = 0.5$, $\delta = 0$), and (iii) **outcome shift** ($\gamma = 0$, $\alpha_k = 1$, $\delta = 1.5$).



(a) Homogeneous clients

(b) Covariate shift

(c) Outcome shift

(d) FedHeart-Disease dataset

Figure 2: Performance comparison on simulated data and the real-world FedHeart dataset.

Figure 2a reports test MSE on a 30% held-out mixture sample from the client mixture in the homogeneous setting (averaged over 150 draws). FedForest-Quantiles (with or without H) and Federated Histogram (which does not split on H) perform on par with centralized RF, with FedForest-Quantiles showing lower variance. Local learning and local ensembling perform worse due to limited per-client sample sizes; although unbiased here, they underutilize data relative to federated and centralized methods. FedForest-AvgImp is slightly weaker and degrades when including H, but approaches centralized performance with larger B or higher L.

In the covariate-shift setting (Figure 2b), client supports are nearly disjoint, and the target function requires a split between them. This stresses methods relying on local impurity estimates. FedForest-Quantiles and Federated Histogram remain close to centralized RF since they evaluate splits using pooled statistics. In contrast, FedForest-AvgImp fails in this regime: averaging client-wise gains can miss global splits when each client observes only one side of the partition. Allowing AvgImp to split on H mitigates this by acting as a proxy for client identity. Local ensembling is omitted due to extremely high error (≈ 141.2), as models trained per client extrapolate poorly across disjoint supports.

Finally, Figure 2c shows the outcome-shift setting. Allowing splits on H significantly improves centralized RF and FedForest variants, which can adapt to client-specific offsets, whereas baselines restricted to X cannot capture this heterogeneity and perform substantially worse. Federated Histogram, not designed to split on H, is particularly affected in this regime.

Real data. We further validate our approach on the FedHeart Disease dataset from the FLamby federated benchmark [23]. This binary classification task comprises 740 patients and 13 clinical covariates distributed across four institutions. We compare our approach against two standard federated baselines evaluated by [23]: **FedAvg LR**, a logistic regression model trained via Federated Averaging [20], and **FedAvg NN**, a two-layer neural network trained using the same protocol. As shown in Figure 2d, both FedForest Quantiles and FedForest AvgImp achieve strong predictive performance, on par with the tree-based baselines (Federated Histograms and Local Learning). In contrast, the parametric

FedAvg baselines (logistic regression and neural network) perform substantially worse. These results illustrate how flexible tree-based methods are often better suited for heterogeneous tabular datasets.

In Table 1, we evaluate 6 regression and 7 classification public datasets, partitioned across $K = 20$ clients with covariate heterogeneity driven by parameter $\alpha \in \{10, 1, 0.1\}$ (see Section B). Our *FedForest* methods remain competitive with Scikit's centralized RF and neural networks.

Table 1: Performance across real datasets: R^2 for the first six and balanced accuracy for the remaining seven. Bold and underlined values indicate best and second-best reported means, resp.

Dataset	α	Central RF	Central NN	FedForest Quantiles	FedForest AvgImp	Fed. Histogram	Local Ensemble	Local Learning
diabetes	10	0.40 ± 0.06	0.36 ± 0.06	$\mathbf{0.41 \pm 0.07}$	$\mathbf{0.41 \pm 0.05}$	$\mathbf{0.41 \pm 0.05}$	0.33 ± 0.05	0.26 ± 0.09
	1	$\mathbf{0.46 \pm 0.05}$	0.3 ± 0.3	$\mathbf{0.46 \pm 0.05}$	0.44 ± 0.06	0.44 ± 0.06	0.31 ± 0.04	0.24 ± 0.08
	0.1	0.43 ± 0.02	0.3 ± 0.2	0.43 ± 0.04	$\mathbf{0.44 \pm 0.04}$	0.43 ± 0.05	0.21 ± 0.04	0.25 ± 0.02
bodyfat	10	0.92 ± 0.02	0 ± 1	$\mathbf{0.93 \pm 0.02}$	0.92 ± 0.02	0.92 ± 0.02	0.25 ± 0.03	0.46 ± 0.03
	1	0.90 ± 0.02	$\mathbf{0.91 \pm 0.03}$	$\mathbf{0.91 \pm 0.02}$	0.90 ± 0.01	0.90 ± 0.02	0.26 ± 0.07	0.3 ± 0.2
	0.1	0.90 ± 0.02	0.87 ± 0.08	$\mathbf{0.91 \pm 0.02}$	0.89 ± 0.02	$\mathbf{0.91 \pm 0.03}$	0.25 ± 0.06	0.6 ± 0.1
space-ga	10	0.57 ± 0.04	$\mathbf{0.71 \pm 0.03}$	0.58 ± 0.04	0.57 ± 0.04	0.59 ± 0.03	0.30 ± 0.02	0.28 ± 0.02
	1	0.57 ± 0.04	$\mathbf{0.71 \pm 0.03}$	0.58 ± 0.05	0.56 ± 0.04	0.58 ± 0.04	0.28 ± 0.03	0.29 ± 0.02
	0.1	0.56 ± 0.03	$\mathbf{0.71 \pm 0.02}$	0.57 ± 0.03	0.56 ± 0.02	0.57 ± 0.03	0.19 ± 0.03	0.34 ± 0.02
puma8NH	10	0.57 ± 0.01	$\mathbf{0.67 \pm 0.01}$	0.58 ± 0.01	0.548 ± 0.007	0.569 ± 0.010	0.194 ± 0.003	0.190 ± 0.004
	1	0.578 ± 0.007	$\mathbf{0.678 \pm 0.005}$	0.589 ± 0.005	0.563 ± 0.006	0.578 ± 0.006	0.196 ± 0.005	0.206 ± 0.008
	0.1	0.568 ± 0.009	$\mathbf{0.675 \pm 0.008}$	0.581 ± 0.008	0.556 ± 0.008	0.575 ± 0.005	0.155 ± 0.008	0.26 ± 0.02
cpu-act	10	0.971 ± 0.003	$\mathbf{0.977 \pm 0.002}$	0.962 ± 0.003	0.967 ± 0.004	0.975 ± 0.003	0.29 ± 0.02	0.28 ± 0.02
	1	0.972 ± 0.003	$\mathbf{0.976 \pm 0.003}$	0.956 ± 0.003	0.966 ± 0.003	$\mathbf{0.976 \pm 0.002}$	0.29 ± 0.01	0.33 ± 0.04
	0.1	0.973 ± 0.002	$\mathbf{0.978 \pm 0.002}$	0.959 ± 0.001	0.970 ± 0.002	0.976 ± 0.002	0.25 ± 0.02	0.47 ± 0.04
kin8nm	10	0.496 ± 0.007	$\mathbf{0.905 \pm 0.005}$	0.534 ± 0.002	0.501 ± 0.005	0.527 ± 0.007	0.252 ± 0.003	0.245 ± 0.005
	1	0.504 ± 0.006	$\mathbf{0.908 \pm 0.005}$	0.538 ± 0.006	0.506 ± 0.006	0.540 ± 0.008	0.246 ± 0.006	0.253 ± 0.005
	0.1	0.503 ± 0.007	$\mathbf{0.904 \pm 0.007}$	0.535 ± 0.007	0.520 ± 0.008	0.536 ± 0.005	0.20 ± 0.02	0.286 ± 0.009
breast-cancer	10	0.951 ± 0.009	0.92 ± 0.03	$\mathbf{0.96 \pm 0.02}$	$\mathbf{0.96 \pm 0.01}$	0.95 ± 0.02	0.94 ± 0.01	0.90 ± 0.03
	1	$\mathbf{0.95 \pm 0.02}$	0.92 ± 0.05	$\mathbf{0.95 \pm 0.02}$	$\mathbf{0.95 \pm 0.02}$	$\mathbf{0.95 \pm 0.01}$	0.91 ± 0.02	0.90 ± 0.01
	0.1	$\mathbf{0.964 \pm 0.006}$	0.93 ± 0.03	0.960 ± 0.007	0.96 ± 0.01	0.96 ± 0.01	0.90 ± 0.02	0.91 ± 0.02
SPECTF	10	0.81 ± 0.07	0.61 ± 0.02	$\mathbf{0.84 \pm 0.08}$	0.79 ± 0.09	0.83 ± 0.08	0.5000	0.61 ± 0.06
	1	$\mathbf{0.85 \pm 0.03}$	0.63 ± 0.07	0.82 ± 0.04	0.82 ± 0.03	0.83 ± 0.02	0.5000	0.60 ± 0.03
	0.1	0.81 ± 0.05	0.61 ± 0.06	$\mathbf{0.82 \pm 0.07}$	0.78 ± 0.05	0.76 ± 0.04	0.5000	0.63 ± 0.06
Bioresponse	10	0.760 ± 0.009	$\mathbf{0.771 \pm 0.005}$	0.77 ± 0.01	0.753 ± 0.009	0.76 ± 0.01	0.63 ± 0.02	0.63 ± 0.01
	1	0.753 ± 0.009	$\mathbf{0.77 \pm 0.01}$	0.76 ± 0.01	0.75 ± 0.01	0.754 ± 0.008	0.638 ± 0.008	0.631 ± 0.010
	0.1	0.76 ± 0.01	$\mathbf{0.77 \pm 0.02}$	0.765 ± 0.006	0.75 ± 0.01	0.763 ± 0.005	0.62 ± 0.03	0.647 ± 0.008
madelon	10	$\mathbf{0.68 \pm 0.02}$	0.55 ± 0.01	0.679 ± 0.007	0.64 ± 0.02	0.67 ± 0.02	0.59 ± 0.01	0.56 ± 0.02
	1	0.68 ± 0.02	0.54 ± 0.01	$\mathbf{0.69 \pm 0.01}$	0.63 ± 0.01	0.66 ± 0.02	0.59 ± 0.03	0.569 ± 0.008
	0.1	0.67 ± 0.02	0.560 ± 0.009	$\mathbf{0.684 \pm 0.010}$	0.66 ± 0.02	0.66 ± 0.01	0.58 ± 0.05	0.61 ± 0.02
Wine	10	0.98 ± 0.02	0.8 ± 0.1	$\mathbf{0.990 \pm 0.009}$	0.98 ± 0.02	0.98 ± 0.02	0.3333	0.86 ± 0.06
	1	0.97 ± 0.02	0.81 ± 0.10	$\mathbf{0.99 \pm 0.01}$	0.97 ± 0.02	0.97 ± 0.02	0.3333	0.75 ± 0.06
	0.1	0.98 ± 0.02	0.8 ± 0.1	0.98 ± 0.02	0.980 ± 0.007	$\mathbf{0.990 \pm 0.009}$	0.3333	0.81 ± 0.04
Bank-Note	10	0.991 ± 0.007	0.97 ± 0.02	$\mathbf{0.992 \pm 0.007}$	0.989 ± 0.006	$\mathbf{0.992 \pm 0.006}$	0.91 ± 0.02	0.891 ± 0.006
	1	0.988 ± 0.006	0.97 ± 0.01	$\mathbf{0.989 \pm 0.004}$	$\mathbf{0.989 \pm 0.003}$	0.987 ± 0.003	0.89 ± 0.02	0.88 ± 0.02
	0.1	0.990 ± 0.004	0.980 ± 0.007	$\mathbf{0.992 \pm 0.003}$	0.990 ± 0.004	0.989 ± 0.003	0.92 ± 0.02	0.92 ± 0.02
Wisconsin	10	0.959 ± 0.007	0.93 ± 0.04	$\mathbf{0.968 \pm 0.008}$	0.96 ± 0.01	$\mathbf{0.968 \pm 0.007}$	0.937 ± 0.009	0.93 ± 0.02
	1	0.96 ± 0.01	0.95 ± 0.02	$\mathbf{0.972 \pm 0.006}$	$\mathbf{0.972 \pm 0.006}$	0.964 ± 0.008	0.930 ± 0.008	0.91 ± 0.02
	0.1	0.96 ± 0.01	0.96 ± 0.02	$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$	$\mathbf{0.97 \pm 0.01}$	0.87 ± 0.05	0.93 ± 0.03

6 Conclusion

We propose *FedForest*, a federated forest training procedure with formal guarantees, showing that centralized CART and Random Forest behavior can be faithfully recovered from decentralized data without sharing individual samples. By reconstructing global split criteria from local summaries, our approach provides a principled foundation for extending the broader family of CART-based methods—including causal, quantile, and survival forests—to heterogeneous federated settings.

While exchanging only summary statistics reduces direct privacy exposure, these aggregates may still leak sensitive information about local data. Our protocol can be further strengthened with additional safeguards, including regularization (e.g., depth constraints and minimum leaf sizes), adaptive binning ($B_\nu < \min_k n_{\nu,k}$), private quantiles [8], secure aggregation, and client abstention for small local partitions. More broadly, an interesting direction for future work is to extend our framework with formal differential privacy guarantees and to study the resulting privacy–utility trade-offs.

Acknowledgments

The work of Rémi Khellaf is supported by grant ANR-20-THIA-0005 AXIAUM. The work of Aurélien Bellet is supported by grant ANR 22-PECY-0002 IPOP (Interdisciplinary Project on Privacy) project of the Cybersecurity PEPR. The work of Julie Josse is supported by grant ANR-22-PESN-0003 (SMATCH). No competing interest is to be reported.

References

[1] Rodolfo Stoffel Antunes, Cristiano André da Costa, Arne Küderle, Imrana Abdullahi Yari, and Björn Eskofier. Federated learning for healthcare: Systematic review and architecture proposal. *ACM Transactions on Intelligent Systems and Technology (TIST)*, 13(4):1–23, 2022.

[2] Gérard Biau and Erwan Scornet. A random forest guided tour. *Test*, 25(2):197–227, 2016.

[3] Leo Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

[4] Leo Breiman, Jerome H. Friedman, Richard A. Olshen, and Charles J. Stone. *Classification and Regression Trees*. Wadsworth, Monterey, CA, 1984.

[5] Alexandre Cotorobai, Jorge Miguel Silva, and José Luis Oliveira. A federated random forest solution for secure distributed machine learning. *arXiv preprint arXiv:2505.08085*, 2025.

[6] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, and Dinani Amorim. Do we need hundreds of classifiers to solve real world classification problems? *The journal of machine learning research*, 15(1):3133–3181, 2014.

[7] Walter D Fisher. On grouping for maximum homogeneity. *Journal of the American statistical Association*, 53(284):789–798, 1958.

[8] Jennifer Gillenwater, Matthew Joseph, and Alex Kulesza. Differentially private quantiles. In *International Conference on Machine Learning*, pages 3713–3722. PMLR, 2021.

[9] Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? In Sanmi Koyejo, S. Mohamed, A. Agarwal, Danielle Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*, 2022.

[10] Tian Gu, Phil H. Lee, and Rui Duan. COMMUTE: Communication-efficient transfer learning for multi-site risk prediction. *Journal of Biomedical Informatics*, 137:104243, 2023.

[11] Anne-Christin Hauschild, Marta Lemanczyk, Julian Matschinske, Tobias Frisch, Olga Zolotareva, Andreas Holzinger, Jan Baumbach, and Dominik Heid²er. Federated random forests can improve local performance of predictive models for various healthcare applications. *Bioinformatics*, 38(8):2278–2286, 2022.

[12] Hemant Ishwaran, Udaya B Kogalur, Eugene H Blackstone, and Michael S Lauer. Random survival forests. 2008.

[13] Kaggle. 2022 Kaggle Machine Learning & Data Science Survey, 2022.

[14] Peter Kairouz, H Brendan McMahan, Brendan Avent, Aurélien Bellet, Mehdi Bennis, Arjun Nitin Bhagoji, Kallista Bonawitz, Zachary Charles, Graham Cormode, Rachel Cummings, et al. Advances and open problems in federated learning. *Foundations and trends® in machine learning*, 14(1–2):1–210, 2021.

[15] Saikishore Kalloori and Severin Klingler. Cross-silo federated learning based decision trees. In *Proceedings of the 37th ACM/SIGAPP Symposium on Applied Computing*, pages 1117–1124, 2022.

[16] Guolin Ke, Qi Meng, Thomas Finley, Taifeng Wang, Wei Chen, Weidong Ma, Qiwei Ye, and Tie-Yan Liu. Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30, 2017.

[17] Qinbin Li, Zhaomin Wu, Yanzheng Cai, Ching Man Yung, Tianyuan Fu, and Bingsheng He. FedTree: A federated learning system for trees. In *Proceedings of Machine Learning and Systems*, volume 5, 2023.

[18] Songfeng Liu, Jinyan Wang, and Wenliang Zhang. Federated personalized random forest for human activity recognition. *Mathematical Biosciences and Engineering*, 19(1):953–971, 2022.

[19] Yang Liu, Yingting Liu, Zhijie Liu, Yuxuan Liang, Chuishi Meng, Junbo Zhang, and Yu Zheng. Federated forest. *IEEE Transactions on Big Data*, 8(3):843–854, 2020.

[20] Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. In *Artificial intelligence and statistics*, pages 1273–1282. PMLR, 2017.

[21] Nicolai Meinshausen and Greg Ridgeway. Quantile regression forests. *Journal of machine learning research*, 7(6), 2006.

[22] Dinh C Nguyen, Quoc-Viet Pham, Pubudu N Pathirana, Ming Ding, Aruna Seneviratne, Zihuai Lin, Octavia Dobre, and Won-Joo Hwang. Federated learning for smart healthcare: A survey. *ACM Computing Surveys (Csur)*, 55(3):1–37, 2022.

[23] Jean Ogier du Terrail, Samy-Safwan Ayed, Edwige Cyffers, Felix Grimberg, Chaoyang He, Regis Loeb, Paul Mangold, Tanguy Marchand, Othmane Marfoq, and Erum Mushtaq. Flamby: Datasets and benchmarks for cross-silo federated learning in realistic healthcare settings. *Advances in Neural Information Processing Systems*, 35:5315–5334, 2022.

[24] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

[25] Nicola Rieke, Jonny Hancox, Wenqi Li, Fausto Milletari, Holger R Roth, Shadi Albarqouni, Spyridon Bakas, Mathieu N Galtier, Bennett A Landman, Klaus Maier-Hein, et al. The future of digital health with federated learning. *NPJ digital medicine*, 3(1):119, 2020.

[26] Holger R. Roth, Yan Cheng, Yuhong Wen, Isaac Yang, Ziyue Xu, Yuan-Ting Hsieh, Kristopher Kersten, Ahmed Harouni, Can Zhao, Kevin Lu, et al. Nvidia flare: Federated learning from simulation to real-world. *arXiv preprint arXiv:2210.13291*, 2022.

[27] Erwan Scornet and Giles Hooker. Theory of random forests. *Annual Review of Statistics and Its Application*, 13(1):99–121, 2026.

[28] Shahadat Uddin and Haohui Lu. Confirming the statistically significant superiority of tree-based machine learning algorithms over their counterparts for tabular data. *PLOS ONE*, 19(4):1–12, 04 2024.

[29] Stefan Wager and Susan Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242, 2018.

[30] Zijun Wang and Keke Gai. Decision tree-based federated learning: A survey. *Blockchains*, 2(1):40–60, 2024.

[31] Pengcheng Xiang, Ling Zhou, and Lu Tang. Transfer learning via random forests: A one-shot federated approach. *Computational Statistics & Data Analysis*, 197:107975, 2024.

[32] Jie Xu, Benjamin S Glicksberg, Chang Su, Peter Walker, Jiang Bian, and Fei Wang. Federated learning for healthcare informatics. *Journal of healthcare informatics research*, 5(1):1–19, 2021.

A FedForest Communication and Computation Costs

In this section, we detail the communication and computation complexities of our proposed FedForest protocol and compare them against standard baselines. Let T be the number of trees, M the maximum tree depth, d the number of features, K the number of clients, n_k the number of samples at client k, and n the total number of samples.

Communication per node. We count communication in transmitted scalars for one node ν and one client k, with B representing the quantile levels (sketch size), and S representing the sufficient-statistic size (e.g., $S = 3$ for regression and $S = C$ for C-class classification). All costs below are strictly independent of the node sample sizes $(n_{\nu,k}, n_\nu)$.

- *FedForest (exact gain):* Communication occurs in two steps. (i) **Sketching:** the client sends d univariate quantile sketches of size B, for a cost of dB. (ii) **Evaluation:** for each tested threshold in $\mathcal{T}$, the client returns the left-child sufficient statistics, costing S scalars per candidate. The total cost of FedForest per node is therefore $O(dBS)$.
- *Top-L shortlisting:* If the server evaluates only $|\mathcal{L}_\nu| = L \ll d$ shortlisted features, the evaluation term strictly reduces to $O(LBS)$.
- *AvgImp (Top-L + averaged local gains):* If clients compute their impurity gains locally at the candidate thresholds of the shortlisted features, they send *one scalar per threshold* instead of S. Combining Top-L with AvgImp therefore reduces the evaluation communication to $O(LB)$, at the price of shifting computation from the server to the clients for the local gain evaluation.

Computation complexity. In a centralized Random Forest, all computation is localized at the server, requiring $O(T2^M dn)$ operations to route data and evaluate splits across all nodes. In *FedForest*, this burden is decoupled:

- *Client-side computation:* At each level of the tree, a client must route its local data and update the quantile sketches for the active nodes. Over T trees and M depth levels, this requires $O(TMdn_k)$ operations, scaling only with the client's local dataset size. For the AvgImp variant, clients additionally sort feature importances, adding an $O(T2^M d \log d)$ overhead.
- *Server-side computation:* The server never touches raw data. It solely aggregates K sketches of size B across d features for up to 2^M nodes per tree. This results in a server computation complexity of $O(T2^M dBK)$.

Table 2: Communication and computation costs required for the entire training phase across centralized, local, and federated methods.

Method	Comm.: Client → Server	Comm.: Server → all Clients	Computation: Client	Computation: Server
Centralized RF	N/A	N/A	N/A	$O(T2^M dn)$
FedForest Quantiles	$O(T2^M dB)$	$O(KT2^M dB)$	$O(TMdn_k)$	$O(T2^M dBK)$
FedForest AvgImp Top-L	$O(T2^M(d+LB))$	$O(KT2^M LB)$	$O(TMdn_k + T2^M d\log d)$	$O(KT2^M(d+LB))$
Federated Histograms	$O(T2^M dB_{\text{hist}})$	$O(KT2^M)$	$O(TMdn_k)$	$O(T2^M dB_{\text{hist}}K)$
Local Ensembling	$O(T2^M)$	N/A	$O(TMdn_k)$	N/A
Parametric (FedAvg)	$O(T_{\text{comm}}P)$	$O(KT_{\text{comm}}P)$	$O(T_{\text{comm}}Pn_k)$	$O(T_{\text{comm}}KP)$

Table 2 highlights several scalability properties of federated tree training. First, the communication cost of *FedForest* is independent of the total sample size n: once local summaries have been computed, communication depends only on the forest structure (T, M), the feature dimension d, and the sketch resolution B. Second, Top-L shortlisting reduces the number of feature–threshold pairs exchanged and evaluated. In high-dimensional regimes with $L \ll d$, this substantially lowers the server-to-client broadcast cost, from $O(KT2^M dB)$ to $O(KT2^M LB)$ for the evaluation stage. Third, the computational load is naturally distributed: clients perform the data-dependent operations on their local samples, while the server only aggregates sketches and sufficient statistics. Unlike *FedAvg*, which requires an undetermined number of communication rounds (T_{comm}) to converge, *FedForest* proceeds level by level and therefore requires a deterministic $O(M)$ synchronized rounds for trees of depth at most M.

B Benchmark on Public Datasets

We give further details on the evaluation in Table 1 of the methods over 13 public tabular datasets, comprising 6 regression and 7 classification tasks from OpenML and `scikit-learn`. Since these datasets are not naturally multicentric, we simulate a federated setting by partitioning each dataset across $K = 20$ clients using a Dirichlet allocation. The concentration parameter α controls the degree of heterogeneity: small values, such as $\alpha = 0.1$, induce highly non-*i.i.d.* partitions with pronounced covariate shift, while larger values yield increasingly homogeneous client distributions.

We compare FedForest against the tree-based baselines considered in the main text, as well as a federated neural-network baseline with two hidden layers of sizes 128 and 64.

As α decreases, client distributions become more fragmented and local methods degrade accordingly. In contrast, *FedForest Quantiles* and *FedForest AvgImp* remain stable under covariate shift and closely track the centralized scikit-learn Random Forest baseline. This supports our claim that aggregating split information across clients is substantially more effective than training isolated local trees, even when those local predictors are later ensembled.

Across the benchmark, the proposed FedForest variants consistently rank among the best-performing methods. *FedForest Quantiles* is typically a top performer across tasks and heterogeneity regimes, while *FedForest AvgImp* achieves comparable performance in many cases. This suggests that averaging local impurity gains is an effective practical approximation, despite its stronger theoretical guarantees being restricted to homogeneous settings.

Tree-based federated methods also compare favorably with the neural-network baseline, especially in more heterogeneous regimes where flexible nonparametric splits appear better adapted to the data structure. Federated Histograms remain competitive, but *FedForest Quantiles* often matches or improves upon them, supporting the use of quantile-based candidates rather than fixed-width bins.

Performance in fragmented settings To further evaluate the robustness of our approaches in highly fragmented scenarios, we conducted additional experiments on the Banknote Authentication dataset. We varied the number of clients from $K = 1$ (centralized) up to $K = 100$, maintaining the same base experimental setup.

As shown in Table 3, **FedForest remains highly stable as fragmentation increases**. The performance of both FedForest variants stays remarkably close to the Centralized Scikit RF baseline across all values of K. In stark contrast, Local Learning and Federated Ensemble approaches degrade sharply as the data becomes divided into smaller, isolated client pools.

These results demonstrate that the primary difficulty in fragmented settings is not the quantile construction itself, but rather the inherent failure of purely local approaches to recover the global split structure when individual clients lack sufficient data.

Table 3: Predictive performance on the Banknote Authentication dataset across varying degrees of client fragmentation ($K = 1$ to $K = 100$). Values indicate mean $\pm$ standard deviation.

Method	K=1	K=2	K=5	K=10	K=20	K=50	K=100
Centralized Scikit RF	0.985 ± 0.007	0.991 ± 0.002	0.986 ± 0.009	0.987 ± 0.005	0.991 ± 0.006	0.990 ± 0.004	0.991 ± 0.001
Fed. Ensemble (Hauschild et al.)	0.988 ± 0.005	0.986 ± 0.005	0.96 ± 0.01	0.93 ± 0.01	0.91 ± 0.01	0.87 ± 0.02	0.5000
Fed. Histogram (Kalloori et al.)	0.996 ± 0.001	0.998 ± 0.004	0.996 ± 0.003	0.994 ± 0.004	0.994 ± 0.005	0.992 ± 0.007	0.997 ± 0.001
FedForest AvgImp (ours)	0.994 ± 0.002	0.996 ± 0.004	0.993 ± 0.006	0.991 ± 0.004	0.996 ± 0.001	0.988 ± 0.003	0.989 ± 0.003
FedForest Quantiles (ours)	0.988 ± 0.008	0.993 ± 0.001	0.986 ± 0.007	0.990 ± 0.005	0.9953 ± 0.0006	0.991 ± 0.007	0.993 ± 0.002
Local Learning	0.988 ± 0.005	0.980 ± 0.005	0.949 ± 0.005	0.913 ± 0.007	0.89 ± 0.01	0.79 ± 0.03	0.55 ± 0.05

C Proofs

C.1 Candidate thresholds with quantile sketching

Proof of Theorem 3.1. The proof proceeds in first establishing the error bound for local approximations, and then extending it to the global aggregate.

Step 1: Local Error Bound. Consider a client k transmitting B exact quantiles $q_0, \dots, q_B$ corresponding to ranks $0, 1/B, \dots, 1$ of a given feature (we omit the feature index j in the notation) at node ν. $\forall x \in [q_i, q_{i+1}]$, the monotonicity of the CDF implies that the true rank $F_{\nu,k}(x) \in [i/B, (i+1)/B]$. Similarly, the linear interpolant $\hat{F}_{\nu,k}(x)$ constructed by the server is also bounded strictly within $[i/B, (i+1)/B]$. Since both the true function and the approximation are in the same interval of height $1/B$, their absolute difference is bounded by that height:

$$\left|\hat{F}_{\nu,k}(x) - F_{\nu,k}(x)\right| \leq \frac{i+1}{B} - \frac{i}{B} = \frac{1}{B}.$$

Step 2: Global Aggregation. We now evaluate the error of the global reconstructed CDF $\tilde{F}_\nu(x) = \sum w_k \hat{F}_{\nu,k}(x)$ relative to the true global CDF $F_\nu(x) = \sum w_k F_{\nu,k}(x)$, where $w_k = n_{\nu,k}/n_\nu$:

$$\begin{aligned}
|\tilde{F}_\nu(x) - F_\nu(x)| &= \left|\sum_{k=1}^{K} w_k \left(\hat{F}_{\nu,k}(x) - F_{\nu,k}(x)\right)\right| \\
&\leq \sum_{k=1}^{K} w_k \left|\hat{F}_{\nu,k}(x) - F_{\nu,k}(x)\right| \quad \text{(Triangle Inequality)} \\
&\leq \sum_{k=1}^{K} w_k \left(\frac{1}{B}\right) \\
&= \frac{1}{B} \sum_{k=1}^{K} w_k = \frac{1}{B}.
\end{aligned}$$

Thus, the global rank error is strictly bounded by $\epsilon = 1/B$, independent of the heterogeneity of the local distributions $F_{\nu,k}$.

□

Proof of Corollary 3.2. We consider a centralized candidate $(j, t_{\text{cent}}) \in \mathcal{T}_{\text{cent}}$. We write F_ν and $\tilde{F}_\nu$ for the pooled empirical CDF and its reconstruction (as in Theorem 3.1). By construction, the federated candidate set for this feature is

$$\mathcal{T}_\nu^{(j)} = \left\{\tilde{F}_\nu^{-1}\left(\tfrac{b}{B}\right) : \; b = 1, \dots, B-1\right\}.$$

Hence there exists $t_{\text{fed}} \in \mathcal{T}_\nu^{(j)}$ such that

$$\left|\tilde{F}_\nu(t_{\text{fed}}) - F_\nu(t_{\text{cent}}\right| \leq \frac{1}{2B},$$

since the grid points $\{b/B\}$ are spaced by $1/B$. We now bound the induced disagreement in left-child assignments:

$$\frac{1}{n_\nu} \sum_{i \in \mathcal{S}_\nu} \mathbb{1}(\mathbb{1}(x_i \leq t_{\text{cent}}) \neq \mathbb{1}(x_i \leq t_{\text{fed}})) = \left|F_\nu(t_{\text{fed}}) - F_\nu(t_{\text{cent}})\right|.$$

By the triangle inequality and Theorem 3.1,

$$\left|F_\nu(t_{\text{fed}}) - F_\nu(t_{\text{cent}}\right| \leq \left|F_\nu(t_{\text{fed}}) - \tilde{F}_\nu(t_{\text{fed}})\right| + \left|\tilde{F}_\nu(t_{\text{fed}}) - F_\nu(t_{\text{cent}}\right| \leq \frac{1}{B} + \frac{1}{2B} = \frac{3}{2B}.$$

□

C.2 AvgImp impurity reduction estimator: homogeneous setting

Proof of Theorem 3.3. By Equation (5), the approximation error equals the split heterogeneity correction exactly:

$$\left|\Delta I(j,t;\mathcal{S}_\nu) - \widehat{\Delta I}_{\mathrm{AvgImp}}(j,t;\mathcal{S}_\nu)\right| = |\Delta\mathcal{E}(j,t;\nu)|,$$

with $\Delta\mathcal{E}(j,t;\nu) := \mathcal{E}(\nu) - \frac{n_L}{n_\nu}\mathcal{E}(\nu_L) - \frac{n_R}{n_\nu}\mathcal{E}(\nu_R)$, so it suffices to show $\mathcal{E}(\nu) = O_p(K/n_\nu)$ for any node ν.

Regression (variance impurity). Let $\mu_{\nu,k} := \frac{1}{n_{\nu,k}}\sum_{i\in\mathcal{S}_{\nu,k}} Y_i$ and $\mu_\nu := \frac{1}{n_\nu}\sum_{i\in\mathcal{S}_\nu} Y_i$ denote the local and global empirical means at node ν, respectively. The node heterogeneity is $\mathcal{E}(\nu) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu}(\mu_{\nu,k}-\mu_\nu)^2 \geq 0$. Under Assumption 2.1, each $\mu_{\nu,k}$ is an unbiased estimator of the population mean $\mu := \mathbb{E}[Y]$, and so is μ_ν, giving $\mathbb{E}[\mu_{\nu,k}-\mu_\nu] = 0$ and $\mathrm{Var}(\mu_{\nu,k}) = O(1/n_{\nu,k})$ (finite by $\mathbb{E}[Y^2] < \infty$). Taking expectations:

$$\mathbb{E}\big[\mathcal{E}(\nu)\big] = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu}\,\mathbb{E}\big[(\mu_{\nu,k}-\mu_\nu)^2\big] \leq \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu}\,\mathrm{Var}(\mu_{\nu,k}) = \sum_{k=1}^K O\!\left(\frac{1}{n_\nu}\right) = O\!\left(\frac{K}{n_\nu}\right),$$

where we used $\mathbb{E}[(\mu_{\nu,k}-\mu_\nu)^2] \leq 2\,\mathrm{Var}(\mu_{\nu,k}) + 2\,\mathrm{Var}(\mu_\nu) = O(1/n_{\nu,k})$. Since $\mathcal{E}(\nu) \geq 0$, Markov's inequality gives $\mathcal{E}(\nu) = O_p(K/n_\nu)$.

Classification (Gini and entropy impurities). Let $\mathbf{p}_{\nu,k} := \big(p_{\nu,k}^{(c)}\big)_{c=1}^C$ and $\mathbf{p}_\nu := \big(p_\nu^{(c)}\big)_{c=1}^C$ be the local and global empirical class-probability vectors at node ν, where $p_{\nu,k}^{(c)} := \frac{1}{n_{\nu,k}}\sum_{i\in\mathcal{S}_{\nu,k}} \mathbf{1}[Y_i = c]$ and $p_\nu^{(c)} := \frac{1}{n_\nu}\sum_{i\in\mathcal{S}_\nu}\mathbf{1}[Y_i = c]$ for each class $c \in \{1,\dots,C\}$.

For the *Gini* index, $\mathcal{E}_{\mathrm{Gini}}(\nu) = \sum_k \frac{n_{\nu,k}}{n_\nu}\|\mathbf{p}_{\nu,k}-\mathbf{p}_\nu\|^2$, and since $\mathbb{E}\|\mathbf{p}_{\nu,k}-\mathbf{p}_\nu\|^2 = O(1/n_{\nu,k})$ by the same moment argument, $\mathcal{E}_{\mathrm{Gini}}(\nu) = O_p(K/n_\nu)$.

For *entropy*, $\mathcal{E}_{\mathrm{Entropy}}(\nu)$ equals the Jensen–Shannon divergence

$$\mathrm{JSD}\big(\{\mathbf{p}_{\nu,k}\}\,\|\,\mathbf{p}_\nu\big) := \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} D_{\mathrm{KL}}\big(\mathbf{p}_{\nu,k}\,\|\,\mathbf{p}_\nu\big), \qquad D_{\mathrm{KL}}(\mathbf{p}\,\|\,\mathbf{q}) := \sum_{c=1}^C p^{(c)}\log\frac{p^{(c)}}{q^{(c)}}.$$

For each k, a second-order Taylor expansion of $D_{\mathrm{KL}}(\mathbf{p}_{\nu,k}\|\mathbf{p}_\nu)$ around $\mathbf{p}_{\nu,k} = \mathbf{p}_\nu$ gives

$$D_{\mathrm{KL}}\big(\mathbf{p}_{\nu,k}\,\|\,\mathbf{p}_\nu\big) = \frac{1}{2}\sum_{c=1}^C \frac{(p_{\nu,k}^{(c)} - p_\nu^{(c)})^2}{p_\nu^{(c)}} + O\big(\|\mathbf{p}_{\nu,k}-\mathbf{p}_\nu\|^3\big) = O\big(\|\mathbf{p}_{\nu,k}-\mathbf{p}_\nu\|^2\big),$$

where the last equality holds provided $p_\nu^{(c)} \geq \epsilon > 0$ for all c (so the denominator is bounded away from zero). Hence $\mathcal{E}_{\mathrm{Entropy}}(\nu) = O_p(K/n_\nu)$ by the same moment argument as for Gini.

Conclusion. The same bound applies to the child nodes ν_L and ν_R (which each receive a $\Theta(n_\nu)$ fraction of samples under a balanced-split regularity condition), so $|\Delta\mathcal{E}(j,t;\nu)| \leq \mathcal{E}(\nu) + \frac{n_L}{n_\nu}\mathcal{E}(\nu_L) + \frac{n_R}{n_\nu}\mathcal{E}(\nu_R) = O_p(K/n_\nu)$.

□

C.3 Impurity Decomposition

Proof of Theorem 3.4 which states:

$$\Delta I(j,t;\mathcal{S}_\nu) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu}\Delta I(j,t;\mathcal{S}_{\nu,k}) + \Delta\mathcal{E}(j,t;\nu),$$

where the split-dependent heterogeneity term $\Delta\mathcal{E}(j,t;\nu)$ represents the change in the baseline node heterogeneity $\mathcal{E}(\nu)$ induced by the split (j,t):

$$\Delta\mathcal{E}(j,t;\nu) = \mathcal{E}(\nu) - \left(\frac{n_L(j,t)}{n_\nu}\mathcal{E}(\nu_L(j,t)) + \frac{n_R(j,t)}{n_\nu}\mathcal{E}(\nu_R(j,t))\right).$$

The baseline node heterogeneity $\mathcal{E}(\nu)$ term is defined as the Jensen gap of the local empirical target distributions:

$$\mathcal{E}(\nu) = \Psi\left(\sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \hat{P}_k\right) - \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \Psi(\hat{P}_k),$$

where $\hat{P}_k$ is the empirical distribution of client k's outcome at the node, and Ψ is a standard statistical impurity function (e.g., variance, Gini index, or entropy–see Table 4).

For these split criteria, $\mathcal{E}(\nu) \geq 0$, with $\mathcal{E}(\nu) = 0$ if and only if the local empirical outcome means are equal for all clients.

Proof. We first establish the decomposition of the empirical impurity $I(\mathcal{S}_\nu)$ at a single node ν. Let $\mathcal{S}_\nu$ be the set of samples at node ν, and $\mathcal{S}_{\nu,k}$ the subset of samples belonging to client k, such that $\mathcal{S}_\nu = \bigcup_{k=1}^{K} \mathcal{S}_{\nu,k}$ and $\mathcal{S}_{\nu,k} \cap \mathcal{S}_{\nu,k'} = \emptyset$ for $k \neq k'$. Let P_ν denote the empirical distribution of the target variable Y over the pooled sample $\mathcal{S}_\nu$, and $P_{\nu,k}$ denote the empirical distribution over the local sample $\mathcal{S}_{\nu,k}$. The empirical distribution assigns mass $1/n$ to each observation. Since the datasets are disjoint, the global empirical measure P_ν is the weighted average of the local measures. For any event E:

$$\begin{aligned} P_\nu(E) &= \frac{1}{n_\nu} \sum_{i \in \mathcal{S}_\nu} \mathbb{1}(y_i \in E) \\ &= \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \left(\frac{1}{n_{\nu,k}} \sum_{i \in \mathcal{S}_{\nu,k}} \mathbb{1}(y_i \in E) \right) \\ &= \sum_{k=1}^{K} w_{\nu,k} P_{\nu,k}(E), \end{aligned}$$

where $w_{\nu,k} = \frac{n_{\nu,k}}{n_\nu}$ represents the mixture weight of client k.

The global empirical impurity is defined as a function Ψ (mean squared error, Gini, cross-entropy, etc.) applied to the global distribution P_ν:

$$I(\mathcal{S}_\nu) = \Psi(P_\nu) = \Psi\left(\sum_{k=1}^{K} w_{\nu,k} P_{\nu,k}\right).$$

We add and subtract the weighted sum of local impurities $\sum_{k=1}^{K} w_{\nu,k} \Psi(P_{\nu,k})$:

$$\begin{aligned} I(\mathcal{S}_\nu) &= \underbrace{\sum_{k=1}^{K} w_{\nu,k} \Psi(P_{\nu,k})}_{\text{Weighted Local Impurities}} + \underbrace{\left[\Psi\left(\sum_{k=1}^{K} w_{\nu,k} P_{\nu,k}\right) - \sum_{k=1}^{K} w_{\nu,k} \Psi(P_{\nu,k}) \right]}_{\text{Jensen-gap / Heterogeneity Term } \mathcal{E}(\nu)} \\ &= \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} I(\mathcal{S}_{\nu,k}) + \mathcal{E}(\nu). \end{aligned}$$

Next, we apply this single-node decomposition to the impurity reduction $\Delta I(j, t; \mathcal{S}_\nu)$ of a candidate split (j, t), which produces left and right children ν_L and ν_R of sizes n_L and n_R. By definition:

$$\begin{aligned} \Delta I(j, t; \mathcal{S}_\nu) &= I(\mathcal{S}_\nu) - \frac{n_L}{n_\nu} I(\mathcal{S}_L) - \frac{n_R}{n_\nu} I(\mathcal{S}_R) \\ &= \left(\sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} I(\mathcal{S}_{\nu,k}) + \mathcal{E}(\nu) \right) \\ &\quad - \frac{n_L}{n_\nu} \left(\sum_{k=1}^{K} \frac{n_{L,k}}{n_L} I(\mathcal{S}_{L,k}) + \mathcal{E}(\nu_L) \right) \\ &\quad - \frac{n_R}{n_\nu} \left(\sum_{k=1}^{K} \frac{n_{R,k}}{n_R} I(\mathcal{S}_{R,k}) + \mathcal{E}(\nu_R) \right). \end{aligned}$$

Rearranging the terms and noting that $\frac{n_L}{n_\nu}\frac{n_{L,k}}{n_L} = \frac{n_{L,k}}{n_\nu} = \frac{n_{\nu,k}}{n_\nu}\frac{n_{L,k}}{n_{\nu,k}}$ (and similarly for the right child), we can group the local impurities for each client:

$$\Delta I(j,t;\mathcal{S}_\nu) = \sum_{k=1}^{K} \frac{n_{\nu,k}}{n_\nu} \underbrace{\left(I(\mathcal{S}_{\nu,k}) - \frac{n_{L,k}}{n_{\nu,k}} I(\mathcal{S}_{L,k}) - \frac{n_{R,k}}{n_{\nu,k}} I(\mathcal{S}_{R,k}) \right)}_{\Delta I(j,t;\mathcal{S}_{\nu,k})}$$
$$+ \underbrace{\mathcal{E}(\nu) - \left(\frac{n_L}{n_\nu}\mathcal{E}(\nu_L) + \frac{n_R}{n_\nu}\mathcal{E}(\nu_R) \right)}_{\Delta\mathcal{E}(j,t;\nu)}.$$

This yields the decomposition in (5).

Finally, we show that for strictly concave impurities,

$$\mathcal{E}(\nu) = 0 \iff \forall (k,k'), \hat{\mathbb{E}}[Y \mid \mathcal{S}_\nu, H = k] = \hat{\mathbb{E}}[Y \mid \mathcal{S}_\nu, H = k'].$$

Let $\mu_{\nu,k} = \hat{\mathbb{E}}[Y \mid \mathcal{S}_\nu, H = k]$ denote the local empirical mean outcome (regression) or class probability vector (classification).

For regression, the Law of Total Variance identifies $\mathcal{E}(\nu)$ as the variance of these local means: $\mathcal{E}(\nu) = \sum \frac{n_{\nu,k}}{n_\nu}(\mu_{\nu,k} - \mu_\nu)^2$.

For classification, $\mu_{\nu,k}$ is the vector of class probabilities (since $\hat{\mathbb{E}}[Y_{\text{one-hot}}] = \mathbf{p}$), and impurity is defined as $\Psi(\mu_{\nu,k})$ where Ψ is strictly concave (e.g., Gini or Entropy). By Jensen's inequality, the gap $\mathcal{E}(\nu) = \Psi\left(\sum \frac{n_{\nu,k}}{n_\nu}\mu_{\nu,k}\right) - \sum \frac{n_{\nu,k}}{n_\nu}\Psi(\mu_{\nu,k})$ is non-negative.

In both cases, due to strict concavity, $\mathcal{E}(\nu) = 0 \iff \mu_{\nu,1} = \cdots = \mu_{\nu,K}$, i.e., the mean outcome or class distribution is identical across all clients.

□

C.4 Exact Computation of CART Criteria from Local Statistics

Proof of Theorem 3.5. We show that standard CART impurities can be computed exactly from additive client-side sufficient statistics, without centralizing raw samples. Let $\mathcal{S}_{\nu,k}$ be the samples held by client k at node ν, with $n_{\nu,k} = |\mathcal{S}_{\nu,k}|$. Each client reports a statistic vector $\mathbf{s}_{\nu,k}$, and the server forms

$$\mathbf{s}_\nu = \sum_{k=1}^{K} \mathbf{s}_{\nu,k}. \tag{7}$$

The impurity at node ν is then obtained by a deterministic map $\Psi(\mathbf{s}_\nu)$, which we now detail for the following standard CART criteria.

Regression: MSE. Each client reports

$$\mathbf{s}_{\nu,k} = \left(n_{\nu,k}, \sum_{i \in \mathcal{S}_{\nu,k}} y_i, \sum_{i \in \mathcal{S}_{\nu,k}} y_i^2 \right). \tag{8}$$

After aggregation, write $\mathbf{s}_\nu = (n_\nu, S_\nu, Q_\nu)$. The MSE impurity is

$$\Psi(\mathbf{s}_\nu) = \frac{Q_\nu}{n_\nu} - \left(\frac{S_\nu}{n_\nu} \right)^2. \tag{9}$$

Classification: Gini and entropy. For C classes, each client reports class counts

$$\mathbf{s}_{\nu,k} = \left(n_{\nu,k}, \sum_{i \in \mathcal{S}_{\nu,k}} \mathbb{1}(y_i = 1), \ldots, \sum_{i \in \mathcal{S}_{\nu,k}} \mathbb{1}(y_i = C) \right). \tag{10}$$

After aggregation, write $\mathbf{s}_\nu = (n_\nu, N_{\nu,1}, \ldots, N_{\nu,C})$. The Gini impurity is

$$\Psi_{\mathrm{Gini}}(\mathbf{s}_\nu) = 1 - \sum_{c=1}^{C} \left(\frac{N_{\nu,c}}{n_\nu} \right)^2, \tag{11}$$

and the Shannon entropy impurity is

$$\Psi_{\mathrm{Ent}}(\mathbf{s}_\nu) = -\sum_{c=1}^{C} \frac{N_{\nu,c}}{n_\nu} \log_2 \left(\frac{N_{\nu,c}}{n_\nu} \right), \tag{12}$$

with zero-count terms omitted.

Thus, for MSE, Gini, and entropy, the global CART impurity is a deterministic function of summed local statistics. Therefore, split gains can be evaluated exactly from federated aggregates alone. □

C.5 Top-L maximum gains aggregation recovers the true optimal feature

Proposition C.1 (Screening consistency under i.i.d. clients)**.** *Consider a node A and let $\Delta\mathcal{I}_\nu(j)$ denote the population (pooled) best gain for feature j. Let j^* be the unique maximizer of $\Delta\mathcal{I}_\nu(\cdot)$ with margin $\gamma := \Delta\mathcal{I}_\nu(j^*) - \max_{j \neq j^*} \Delta\mathcal{I}_\nu(j) > 0$.*

Assume that for client k, the empirical best-gain scores $M_{\nu,k}(j)$ concentrate around their population means with exponential tails (i.e., satisfy a sub-Gaussian tail bound). Then, there exist constants $C_1, C_2 > 0$ such that the probability of the optimal feature j^* falling out of the top-L local ranking decays exponentially with the local sample size $n_{\nu,k}$:

$$\mathbb{P}(\mathrm{rank}_k(j^*) > L) \leq C_1\, d\, \exp(-C_2\, n_{\nu,k}\, \gamma^2). \tag{13}$$

Proof. The proof relies on decomposing the ranking failure into pairwise estimation errors and applying concentration bounds.

For $L \geq 1$, the event $\{\mathrm{rank}_k(j^*) > L\}$ implies that the optimal feature j^* was empirically outperformed by at least L other features, so $\{\mathrm{rank}_k(j^*) > L\} \subseteq \cdots \subseteq \{\mathrm{rank}_k(j^*) > 1\}$, so we can bound the probability of the optimal feature not to be selected by:

$$\mathbb{P}(\mathrm{rank}_k(j^*) > L) \leq \mathbb{P}\left(\bigcup_{j \neq j^*} \{M_{\nu,k}(j) \geq M_{\nu,k}(j^*)\} \right).$$

Now consider a specific suboptimal feature $j \neq j^*$. By definition, the true gain gap is $\Delta\mathcal{I}_\nu(j^*) - \Delta\mathcal{I}_\nu(j) \geq \gamma$. For the empirical scores to reverse this order, i.e. satisfying $M_{\nu,k}(j) \geq M_{\nu,k}(j^*)$), their summed estimation errors for features j and j^* must exceed the margin γ. Specifically, let $\varepsilon_k(\cdot) = M_{\nu,k}(\cdot) - \Delta I_\nu(\cdot)$ be the estimation noise. The reversal condition implies:

$$\varepsilon_k(j) - \varepsilon_k(j^*) \geq \Delta I_\nu(j^*) - \Delta I_\nu(j) \geq \gamma.$$

Finally, we have by the Triangle inequality

$$|\varepsilon_k(j)| + |\varepsilon_k(j^*)| \geq |\varepsilon_k(j) - \varepsilon_k(j^*)| \geq \gamma,$$

so for the difference of errors to exceed γ, at least one of the individual error terms must have a magnitude of at least $\gamma/2$. By the concentration assumption (e.g., Hoeffding's inequality for bounded impurity functions), the probability of such a deviation for any single feature decays exponentially with the sample size:

$$\mathbb{P}\left(|\varepsilon_k(j)| \geq \frac{\gamma}{2} \right) \leq c_1 \exp\left(-c_2\, n_{\nu,k} \left(\frac{\gamma}{2} \right)^2 \right).$$

Last, applying the union bound over the two error terms for the pair (j, j^*), and subsequently over all $d-1$ possible suboptimal features j, we obtain:

$$\mathbb{P}(\mathrm{rank}_k(j^*) > L) \leq \sum_{j \neq j^*}^{d} 2c_1 \exp\left(-\frac{c_2}{4} n_{\nu,k} \gamma^2 \right).$$

Simplifying the constants yields the final bound $C_1 d \exp(-C_2 n_{\nu,k} \gamma^2)$.

□

D Impurity Decomposition under Covariate Shift

To evaluate a split under heterogeneity efficiently, the heterogeneity term $\mathcal{E}_\Phi$ defined in Theorem 3.4 forbids the mere aggregation of local impurity gains. While one could compute $\mathcal{E}_\Phi(\mathcal{S}_\nu)$ explicitly and add it to the weighted local impurities, reconstructing the global impurity directly from sufficient statistics is more efficient. For standard impurity criteria, the global impurity $I(\mathcal{S}_\nu)$ dependents only on additive moments of the target variable (e.g., sums of targets or counts of classes). This allows for an efficient one-shot protocol: rather than sharing raw data, clients transmit a compact vector of sufficient statistics $\mathbf{s}_{\nu,k}$.

Because these statistics are linear, the server can reconstruct the global statistics $\mathbf{s}_A$ simply by summing the local vectors: $\mathbf{s}_\nu = \sum_{k=1}^K \mathbf{s}_{\nu,k}$. The server then evaluates the global centralized impurity using $\Psi(\cdot)$:

$$I(\mathcal{S}_\nu) = \Psi(\mathbf{s}_\nu) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} \Psi(\mathbf{s}_{\nu,k}) + \mathcal{E}_\Phi(A).$$

By computing $\Psi(\mathbf{s}_\nu)$ directly from the aggregated sums, the server implicitly captures the heterogeneity correction $\mathcal{E}_\Phi(A)$, ensuring the exact same split decision as a centralized algorithm.

The impurity reduction for a candidate split (j, t) is finally computed as:

$$\Delta I(j, t; \mathcal{S}_\nu) = \Psi(\mathbf{s}_\nu) - \left[\frac{n_L}{n_\nu} \Psi(\mathbf{s}_L) + \frac{n_R}{n_\nu} \Psi(\mathbf{s}_R) \right].$$

Table 4 details the minimal statistics $\mathbf{s}$ and the corresponding functional Ψ for common loss functions.

Table 4: Definitions of impurity and sufficient statistics. The column $I(\mathcal{S})$ shows the standard centralized definition. The function $\Psi(\mathbf{s})$ computes this same value using only the aggregated statistics $\mathbf{s}$, allowing for exact federated evaluation.

Objective	**Centralized Impurity** $I(\mathcal{S}_\nu)$	**Federated Impurity** $\Psi(\mathbf{s}_\nu)$	**Sufficient Statistics** $\mathbf{s}_{\nu,k}$
Regression (MSE)	$\frac{1}{n} \sum_{i \in \mathcal{S}_\nu} (y_i - \bar{y})^2$	$\frac{S_\nu^{(YY)}}{n_\nu} - \left(\frac{S_\nu^{(Y)}}{n_\nu} \right)^2$	$\{n_{\nu,k}, S_{\nu,k}^{(Y)}, S_{\nu,k}^{(YY)}\}$
Classification (Gini)	$1 - \sum_{c=1}^C p_c^2$	$1 - \sum_{c=1}^C \left(\frac{n_{A,c}}{n_\nu} \right)^2$	$\{n_{A,k,c}\}_{c=1}^C$
Classification (Entropy)	$-\sum_{c=1}^C p_c \log p_c$	$-\sum_{c=1}^C \frac{n_c}{n} \log \left(\frac{n_c}{n} \right)$	$\{n_{A,k,c}\}_{c=1}^C$

Standard impurity uses sample mean $\bar{y}$ and class probabilities p_c. In sufficient statistics: $S_{\nu,k}^{(Y)} = \sum_{i=1}^{n_{\nu,k}} y_i$, $S_{\nu,k}^{(YY)} = \sum_{i=1}^{n_{\nu,k}} y_i^2$, and $n_{A,k,c} = \sum_{i=1}^{n_{\nu,k}} \mathbb{1}(y_i = c)$.

Note that the heterogeneity term $\mathcal{E}(A)$ takes explicit forms for each impurity measure:

- **Regression (MSE):** $\mathcal{E}(A) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} (\bar{y}_{\nu,k} - \bar{y}_\nu)^2$
- **Classification (Gini):** $\mathcal{E}(A) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} \sum_{c=1}^C (p_{A,k,c} - p_{A,c})^2$
- **Classification (Entropy):** $\mathcal{E}(A) = \sum_{k=1}^K \frac{n_{\nu,k}}{n_\nu} \sum_{c=1}^C p_{A,k,c} \log \left(\frac{p_{A,k,c}}{p_{A,c}} \right)$.

However, in practice, the server does not need to compute these terms explicitly, as they are accounted for when evaluating the impurity Ψ on the aggregated sufficient statistics.

E Exact Midpoint Enumeration

For completeness, we describe how centralized CART split candidates can be reproduced exactly in a federated setting. At a given node A, centralized CART evaluates midpoints between consecutive

unique feature values. In principle, this can be achieved federatively by having each client transmit its node-level feature values, which the server merges into a globally sorted list.

Formally, for feature j, let $\{v_{j,(r)}\}_{r=1}^{m_{A,j}}$ denote the sorted unique pooled values at node A. The exact candidate set is

$$\mathcal{T}_\nu^{\text{all}} = \bigcup_{j=1}^{d} \left\{ (j,t) : t = \tfrac{v_{j,(r)}+v_{j,(r+1)}}{2},\ 1 \le r < m_{A,j} \right\}.$$

This procedure exactly matches centralized CART but requires transmitting $O(n_\nu d)$ floating-point values per node and produces up to $d(n_\nu - 1)$ candidates. As such, it is impractical beyond small-scale settings and is not used in our method.

F Additional Simulations

Definition of the regression function f. To define a complex, nonlinear, tree-like ground-truth $f : \mathbb{R}^d \to \mathbb{R}$, we distill a tree on an auxiliary dataset sampled over a fixed domain. We draw $N_{\text{aux}} = 10,000$ samples, $x_{\text{aux}} \sim \text{Unif}([-10,10]^d)$ with $d = 20$, and rescale them to $\tilde{x} \in [0,1]^d$. We define the auxiliary target as $y_{\text{aux}} = \sum_{j=1}^{d} \psi_{(j \bmod 3)}(\tilde{x}_j, \tilde{x})$, using three component functions: $\psi_0(z) = \mathbb{1}\{z > 0.5\}$ (step), $\psi_1(z) = \sin(4\pi z)$ (frequency), and $\psi_2(z, \tilde{x}) = \tilde{x}_j \times \tilde{x}_{(j \bmod d)+1}$ (interaction). Finally, we fit a CART regressor of maximum depth 8 on $(x_{\text{aux}}, y_{\text{aux}})$ and define the ground truth $f(\cdot)$ as the resulting outcome function.

F.1 Covariate shift: when local objectives mis-rank global splits

We study two mechanisms by which covariate shift can affect federated split selection. Throughout, *FedForest Quantiles* remains stable because it constructs candidates from a pooled quantile sketch and evaluates them with exact pooled impurities, whereas *FedForest AvgImp* can fail because it optimizes a different objective (the average of local gains).

F.1.1 Disjoint supports

We generate $K = 2$ clients with $n_k = 150$ samples each and features in $\mathbb{R}^d$ (here $d = 5$). Client k draws

$$X_i^{(k)} \sim \mathcal{N}(\mu_k, I_d), \qquad \mu_k = \big((-1)^k \gamma, 0, \ldots, 0\big),$$

so that increasing $\gamma \ge 0$ creates a growing gap between client supports along the first coordinate, effectively moving the pooled data distribution of (X) from i.i.d. samples Assumption 2.1 when $\gamma = 0$ to a (Gaussian here) bimodal mixture with $\rho_1 = \rho_2 = \frac{1}{2}$. Outcomes follow a step rule on $X^{(1)}$:

$$Y_i = f(X_i) + \varepsilon_i, \qquad f(x) = 10 \times \mathbb{1}\{x^{(1)} > 0\}, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2),$$

with $\sigma = 1$, so the irreducible MSE is 1. All RF methods are trained with a maximum depth of 8 and 50 trees. Table 5 shows that as γ increases, FedForest AvgImp fails to select the pooled-data optimal split $(j = 1, t = 0)$, while quantile candidates remain robust and track centralized training.

Table 5: Mean Squared Error (MSE) comparison under Ranking Disagreement (Lower is Better). At higher shift scales ($\delta \ge 2.5$), the local greedy methods (AvgImp) fail to identify the optimal split in the gap between client supports, while global quantile methods remain robust.

Shift (γ)	FedForest AvgImp (X)	FedForest AvgImp (X,H)	FedForest Quantiles (X)	FedForest Quantiles (X,H)	Centralized (X)	Centralized (X,H)
0.0	1.22 ± 0.18	1.22 ± 0.18	1.22 ± 0.22	1.22 ± 0.22	1.22 ± 0.20	1.22 ± 0.21
1.0	1.26 ± 0.22	1.26 ± 0.22	1.20 ± 0.18	1.20 ± 0.18	1.19 ± 0.17	1.19 ± 0.17
2.5	5.86 ± 5.36	5.86 ± 5.36	1.16 ± 0.12	1.16 ± 0.12	1.19 ± 0.17	1.22 ± 0.15
5.0	19.22 ± 3.99	19.22 ± 3.99	1.05 ± 0.04	1.05 ± 0.04	1.06 ± 0.04	1.06 ± 0.04

F.1.2 Overlapping supports: failure driven by the heterogeneity term

Disjoint supports are not required for AvgImp to fail. Here, both clients have overlapping feature support, but the pooled impurity differs from the weighted average of local impurities due to the heterogeneity term $\mathcal{E}(A)$ in Theorem 3.4. We simulate $K = 2$ clients with

$$X_i^{(k)} \sim \mathcal{N}(\mu_k, \sigma_X^2), \qquad \mu_k \in \{-\gamma, +\gamma\}, \qquad \sigma_X = 1.5 I_d,$$

and a shared conditional model that linear in feature $X^{(1)}$:

$$Y_i = X_i^{(1)} + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2).$$

The pooled optimal split is near the pooled population mean (equal to $\mathbb{E}[X] = \frac{1}{2}(-\gamma) + \frac{1}{2}\gamma = 0$ under balanced mixture weights), and every client has data around that point. Nevertheless, as γ grows, $\mathcal{E}(A)$ increases, so maximizing the average local gain can favor thresholds near the client means (around $\pm\gamma$) instead of the pooled optimum. Table 6 confirms this effect: FedForest Quantiles remains close to centralized performance across shifts, while AvgImp degrades as γ increases, despite overlapping supports.

Table 6: Mean Squared Error (MSE) comparison under Ranking Disagreement (Lower is Better). At higher shift scales ($\delta \geq 2.5$), the local greedy methods (AvgImp) fail to identify the optimal split in the gap between client supports, while global quantile methods remain robust.

Shift (γ)	FedForest AvgImp (X)	FedForest AvgImp (X,H)	FedForest Quantiles (X)	FedForest Quantiles (X,H)	Centralized (X)	Centralized (X,H)
0.0	1.22 ± 0.18	1.22 ± 0.18	1.22 ± 0.22	1.22 ± 0.22	1.22 ± 0.20	1.22 ± 0.21
1.0	1.26 ± 0.22	1.26 ± 0.22	1.20 ± 0.18	1.20 ± 0.18	1.19 ± 0.17	1.19 ± 0.17
2.5	5.86 ± 5.36	5.86 ± 5.36	1.16 ± 0.12	1.16 ± 0.12	1.19 ± 0.17	1.22 ± 0.15
5.0	19.22 ± 3.99	19.22 ± 3.99	1.05 ± 0.04	1.05 ± 0.04	1.06 ± 0.04	1.06 ± 0.04

F.2 Covariate and Outcome Shift

We consider the fully heterogeneous regime of Equation (3), where both covariates ($\alpha_k = .5$ and $\gamma_k = 3$) and outcomes ($\delta = 1.5$) vary across clients. Similar hyperparameters are used as in Section 5.

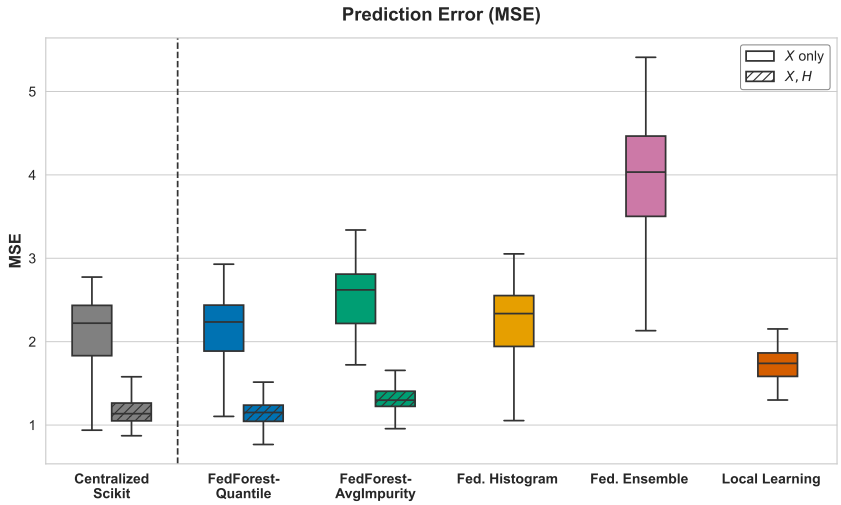


Figure 3: Methods comparison on fully heterogeneous clients (Equation (3)).

The results align with previous conclusions from the covariate shift and outcome shift paragraphs in Section 5.

F.3 Robustness to Outliers

To assess robustness to outliers, we use the same data-generating process as in the homogeneous (*i.i.d.*) simulation setting described in Section 5. We then contaminate the training data by introducing outliers at fractions ranging from 0% to 20% of the total sample size. For the contaminated observations, the corresponding feature vectors are multiplied by a factor of 1000, producing extreme covariate values while leaving the remaining data unchanged.

The results in Table 7 show that the proposed methods are substantially more stable under outlier contamination than the histogram-based federated baseline. In particular, the MSE of Fed. Histogram increases sharply as the outlier fraction grows, rising from 0.51 with no contamination to 1.55 at 20% contamination. By contrast, Fed. Quantiles remains close to the centralized reference across

Table 7: Model Degradation (MSE) vs. Outlier Fraction. Results show the mean ± standard deviation across 20 simulations.

	Outlier Fraction					
Method	**0%**	**2%**	**5%**	**10%**	**15%**	**20%**
Centralized Scikit X	0.48 ± 0.05	0.48 ± 0.05	0.48 ± 0.07	0.52 ± 0.10	0.56 ± 0.09	0.53 ± 0.08
Fed. Average Impurity X (ours)	0.51 ± 0.12	0.55 ± 0.09	0.63 ± 0.08	0.64 ± 0.11	0.65 ± 0.11	0.66 ± 0.09
Fed. Quantiles X (ours)	0.46 ± 0.08	0.48 ± 0.07	0.55 ± 0.08	0.54 ± 0.09	0.59 ± 0.09	0.56 ± 0.08
Fed. Histogram (Kalloori et al.)	0.51 ± 0.11	0.53 ± 0.08	0.67 ± 0.15	0.90 ± 0.18	1.09 ± 0.32	1.55 ± 0.21
Fed. Ensemble (Hauschild et al.)	1.10 ± 0.37	1.08 ± 0.23	1.33 ± 0.24	1.30 ± 0.32	1.47 ± 0.31	1.41 ± 0.17
Local Learning	1.27 ± 0.41	1.22 ± 0.24	1.45 ± 0.27	1.44 ± 0.34	1.61 ± 0.35	1.55 ± 0.21

all contamination levels, with MSE increasing only from 0.46 to 0.56. Fed. Average Impurity also degrades gradually, reaching 0.66 at the highest contamination level.

Histogram-based methods are sensitive to outliers because global min–max binning can stretch the feature range, collapsing most clean samples into coarse bins and reducing split quality. In contrast, quantile-based methods rely on ranks, so extreme values are confined to tail bins while the remaining bins preserve resolution over the clean data. This accounts for their milder degradation under outlier contamination.

F.4 Heterogeneity diagnostics for adaptive optimization

To balance communication efficiency with predictive performance, we run a lightweight diagnostic step at the root node of the first tree. The goal is to decide whether i.i.d.-specific accelerations (top-L screening, local-gain averaging, and client subsampling) are appropriate, or whether we should default to the fully robust mode (exact split evaluation and quantile candidates).

Covariate shift test. We test for covariate shift by assessing the null hypothesis $H_0 : X \perp\!\!\!\perp H$ using a federated discriminator of clients. Instead of estimating density ratios—either parametrically or via kernel methods with high communication cost—we reuse the impurity-based split selection of federated decision trees by treating the client index H as the target variable. A federated Random Forest is trained to predict $H|X$ using standard CART criteria.

Under H_0, the feature distributions $P(X|H = k)$ are identical across clients, implying that no split θ can separate samples by site better than random chance. We quantify this using the site impurity gain. For a (j, t) candidate split at a node ν, each client k sends the count of its local samples falling into the left partitions: $n_{L,k} = \sum_{i \in \mathcal{S}_{\nu,k}} \mathbb{1}(x_i^{(j)} \leq t)$, enough to derive $n_{R,k} = n_{\nu,k} - n_{L,k}$. The server then computes the K-class impurity gain for the site variable H:

$$\Delta I_H(j, t; \mathcal{S}_\nu) = I_H(\mathcal{S}_\nu) - \left(\frac{n_L}{n_\nu} I_H(\mathcal{S}_L) + \frac{n_R}{n_\nu} I_H(\mathcal{S}_R) \right),$$

where $I_H(\mathcal{S}_L) = 1 - \sum_{k=1}^{K} \left(\frac{n_{L,k}}{n_L} \right)^2$ (Gini). Note that although each client contains only a single "class" of site labels–i.e., the local impurity is zero–the aggregation of local counts yields the exact global impurity. We thus reject H_0 whenever the maximal root gain or the predictive performance of the site classifier (e.g., one-vs-rest AUC) exceeds a prescribed threshold. This non-parametric technique is federated and further provides interpretable insights on which features are heterogenously distributed across sites.

Outcome Heterogeneity Test. To detect outcome heterogeneity driven by site effects, one could formally test the null hypothesis that the conditional expectation of the target is invariant across sites. This involves performing $K - 1$ pairwise comparisons (e.g., against a reference site k^*):

$$H_0 : \mathbb{E}[Y \mid X, H = k] = \mathbb{E}[Y \mid X, H = k'].$$

While statistical procedures such as permutation tests, Kruskal-Wallis tests on residuals, or cosine similarity tests can be adapted to the federated setting to provide formal significance levels, they typically incur high communication costs and require delicate threshold calibration. Instead, we propose a pragmatic model-based alternative: we compare the validation performance (e.g., accuracy

or R^2) of two federated Random Forests—one trained solely on features X, and an augmented model trained on (X, H). A significant performance gain by the augmented model implies that the site index H contains predictive signal not captured by X, effectively acting as a proxy for rejecting H_0.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes]

   Justification: We propose a random forest forest training procedure for decentralized data which respects the federated learning constraint, i.e., that does not require sending raw individual data. We provide formal guarantees on its closeness to the centralized baseline.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes]

   Justification: We discuss limitations in the Conclusion section.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [Yes]

Justification: All theoretical claims are proved in the Appendix, and assumptions are made explicit in the theorem formulations.

Guidelines:

- The answer [N/A] means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: The method described in Algorithm 1 is fully reproducible with the information provided in the paper.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
     (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: Code is made available.

Guidelines:

- The answer [N/A] means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: Yes.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
   - The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
   - The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)

- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: The computation costs are available in the appendix.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification:

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: See the Introduction section.

    Guidelines:

    - The answer [N/A] means that there is no societal impact of the work performed.
    - If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
    - Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.

- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
    - If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
    - For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
    - If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [Yes]

    Justification: Yes.

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
    - We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
    - For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

Answer: [Yes]

Justification: Yes.

Guidelines:

- The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
- Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.