PrivateBoost: Rethinking Federated Learning for Patient-Owned Medical Data: Learning from Single Records at Scale

Bernhard Specht[1,2,*] Orhan Ermis[3] Samaher Garbaya[1]
Reinhard Schneider[2] Ricardo Chavarriaga[4] Djamel Khadraoui[3]
Zied Tayeb[1]

[1]MyelinZ
[2]Luxembourg Centre for Systems Biomedicine (LCSB), University of Luxembourg, Esch-sur-Alzette, Luxembourg
[3]Luxembourg Institute of Science and Technology (LIST), Esch-sur-Alzette, Luxembourg
[4]ZHAW Zurich University of Applied Sciences, Winterthur, Switzerland

*Corresponding author: bernhard.specht@myelinz.com

Abstract

The translation of artificial intelligence into clinical practice depends, in large part, on access to data that patients are understandably reluctant to surrender. Federated learning promises to resolve this tension by training models without centralising raw records, yet virtually all deployed systems assume that each participating site holds hundreds or thousands of samples — a condition that hospitals and research consortia may satisfy, but individual patients never will. Here, we show that a fundamentally different regime is both theoretically tractable and practically viable: one in which each participant contributes a single diagnostic record, retains it entirely on their own device, and never communicates with any other participant. We introduce PrivateBoost, a federated gradient boosting framework in which clients distribute Shamir secret shares to a small set of independent shareholders, enabling an aggregator to reconstruct only the aggregate statistics required for tree construction while remaining information-theoretically blind to individual contributions. Crucially, this approach exploits a deep compatibility between gradient boosting and secret sharing: model training depends only on additive aggregates, which align naturally with secure summation, eliminating the need for coordination while maintaining efficiency. A path-hiding extension further prevents shareholders from inferring any patient's traversal of the decision tree, eliminating a side channel that standard protocols leave exposed. Evaluated across four medical datasets encompassing up to 70,692 single-sample clients, PrivateBoost matches centralised XGBoost with identical AUC-ROC (0.818), tolerates up to 80% client dropout per round, and requires fewer than 600 KB of communication per patient over a complete training run — comfortably within the constraints of a mobile device on a standard data plan. These results establish that privacy-preserving collective learning from patient-owned records is not merely a theoretical aspiration but an achievable engineering reality, opening a path towards medical AI that improves with every consenting patient without requiring any of them to share their data.

Keywords: Federated learning, gradient boosting, secret sharing, privacy-preserving machine learning, cross-device learning

1 Introduction

Every patient who consents to medical research faces an implicit bargain that the digital age has rendered increasingly uncomfortable: in exchange for contributing to scientific knowledge, they must relinquish detailed records of their health to institutions whose data governance practices they cannot directly audit Hummel et al. [2021], Montgomery et al. [2021]. This bargain is not incidental to modern medical AI – it is structural. The supervised learning algorithms that underpin diagnostic models require a large quantity of labelled examples, and this quantity demands aggregation, which in turn requires transfer. The result is a landscape in which the most consequential medical datasets are concentrated in the hands of a small number of large institutions, while the patients whose experiences generated that data retain neither custody nor influence Kelly et al. [2019], Vayena et al. [2018]. Federated learning, introduced by McMahan and colleagues in 2016, offered a principled escape from this arrangement McMahan et al. [2017a]. In lieu of moving data to a central model, the model — or at least its gradients — moves to the data McMahan et al. [2017b]. Individual sites train locally and contribute only parameter updates, keeping raw records behind institutional firewalls. The approach has been adopted with genuine enthusiasm across oncology Dayan et al. [2021], cardiology, and rare disease research Rieke et al. [2020], and its privacy properties, while imperfect, represent a meaningful advance over naive data pooling. Yet federated learning as currently practised preserves one assumption so thoroughly that it has become nearly invisible: each participating client holds multiple samples. In the cross-silo paradigm that dominates the literature, clients are hospitals or research institutes, each managing cohorts of hundreds or thousands of patients. This assumption is not merely convenient; it is load-bearing Li et al. [2020a]. Local gradient computation, the engine of model-driven federated learning, produces meaningful statistics only when aggregated over a batch Zhu et al. [2019]. A client holding a single record produces a gradient that is, in a precise sense, the record itself: inspecting it is equivalent to reading the data. The patient as client — an individual contributing their own health record directly from a personal device, without any institutional intermediary, has consequently remained outside the practical reach of federated learning, notwithstanding being, in many respects, its most natural expression. Patients with rare neurological conditions Griggs et al. [2009], individuals enrolled in longitudinal wearable studies, participants in decentralised clinical trials Rogers et al. [2020]: all hold data of genuine scientific value, all face regulatory and personal reasons to withhold it from centralised repositories, and none can satisfy the batch-size assumptions on which existing protocols depend.

We argue that this limitation is not fundamental but architectural, and that addressing it requires moving away from gradient-sharing entirely. The key insight is that gradient boosting, the family of ensemble methods that consistently achieves state-of-the-art performance on the structured tabular data that dominates clinical diagnostics, does not require local computation across batches. It requires aggregate statistics: specifically, sums of gradients and Hessians across all samples falling into each histogram bin. These sums can be computed privately even when each contributor holds a single value, provided the aggregation is mediated by a suitable cryptographic primitive Grinsztajn et al. [2022], Bonawitz et al. [2017a]. Herein, we present PrivateBoost, a federated XGBoost protocol designed for the one-sample-per-client regime. Each patient device computes a single gradient–Hessian pair and distributes it as Shamir secret shares Shamir [1979a] to a fixed, small set of independent shareholders, such as a hospital, a regulatory body, or a cloud provider, none of whom can reconstruct individual values without the cooperation of their peers. Shareholders sum the shares they receive and forward partial sums to an aggregator, which reconstructs only the aggregate statistics needed to determine each tree split. No client communicates with any other client; no party at any point observes an individual patient's contribution. A path-hiding extension, in which clients submit cryptographically indistinguishable shares for every active tree node rather than only their true one, closes

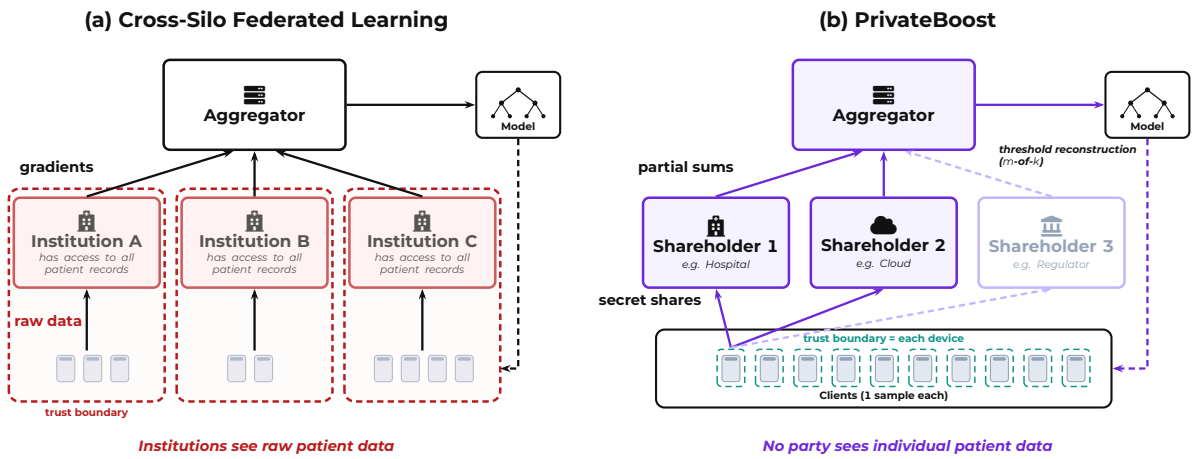

Figure 1: Institution-centric versus patient-level federated learning with PrivateBoost. a, In the conventional cross-silo setting (Institution-centric), patients upload raw data to institutions, which perform local training and share model updates. Each institution has access to all records within its cohort. b, In PrivateBoost, individual patients participate directly: each client holds a single record and distributes secret shares to a small set of independent shareholders. Aggregation is performed only over shared statistics, such that no party—neither shareholders nor aggregator—observes individual patient data.

a residual side channel through which shareholders might otherwise infer a patient's diagnostic trajectory through the decision tree. The protocol provides information-theoretic privacy guarantees, not merely computational ones, meaning that its security does not rest on the presumed hardness of any mathematical problem but on the fundamental properties of polynomial secret sharing. We prove security formally in the simulation paradigm and evaluate the system on four medical datasets spanning up to 70,692 individual clients, demonstrating that PrivateBoost matches centralised XGBoost exactly in predictive performance while remaining robust to the high and unpredictable dropout rates characteristic of real-world mobile deployments Bonawitz et al. [2019]. The vision underpinning this work is of a future where a patient with a single diagnostic record on their phone should be able to contribute to a global model that will help diagnose future patients, without sharing their data with anyone, without coordinating with any other participant, and without trusting any single institution with their privacy. We show that this vision is technically realised. The remainder of this paper is organised as follows. Section 2 situates PrivateBoost within the existing literature on secure aggregation and federated gradient boosting. Section 3 formalises the problem and the three-party architecture. Section 4 describes the protocol in full, including the path-hiding extension. Section 5 establishes the security guarantees through formal proof. Section 6 reports empirical evaluation across four medical datasets spanning up to 70,692 single-sample clients, examining model quality, dropout resilience, and communication cost. Sections 7 and 8 discuss future extensions and conclusions.

2 Related Work

Overall, existing work implicitly assumes multi-sample clients, whereas our setting is orthogonal.

2.1 Secure Aggregation

SecAgg, introduced by Bonawitz and colleagues Bonawitz et al. [2017b], arranges for every pair of clients to agree on a random mask via Diffie-Hellman key exchange; because these masks are constructed to cancel under summation, the server recovers the aggregate without observing

any individual contribution. The cost is topological: each client must perform $O(n)$ pairwise key agreements, and when a client drops out mid-protocol, the remaining participants must coordinate to remove its uncancelled masks. Subsequent work has reduced the per-client communication to polylogarithmic overhead Bell et al. [2020], yet the underlying design still requires each client to exchange keys with a structured neighbourhood of peers — a requirement that sits uneasily with a cross-device setting in which tens of thousands of patients connect intermittently from personal devices and never interact with one another. PrivateBoost sidesteps this constraint entirely: clients distribute shares to a small, fixed set of shareholders in a star topology, with no client-to-client communication at any stage of the protocol.

2.2 Federated Gradient Boosting

Efforts to adapt gradient boosting to federated settings have followed two axes, vertical and horizontal partitioning, but both share a common premise: that each participant holds a substantial local dataset.

In the vertical setting, where parties hold different features for the same samples, SecureBoost Cheng et al. [2021] encrypts gradient statistics with homomorphic encryption so that an active party can determine optimal splits without seeing passive parties' features. Federated Forest Liu et al. [2022] applies a similar feature-partitioned design to random forests. More recently, Xie et al. Xie et al. [2022] extended secret sharing to multi-party vertical federated XGBoost, and FedTree Li et al. [2023] unified both vertical and horizontal settings in a single system with configurable privacy mechanisms. All of these approaches require entity alignment across institutional participants: each party contributes a cohort, never a single record.

The horizontal setting, where parties hold different samples with the same features, is closer to our problem. SimFL Li et al. [2020b] is the most developed approach in this direction: parties exchange locality-sensitive hashes to identify similar samples across institutions, then each party builds trees on its own data using gradients weighted by those cross-party similarities. The mechanism is clever but structurally incompatible with single-sample clients: a party must hold enough local instances to construct a full decision tree, and the similarity lookup degenerates when a party contributes only one record. Moreover, LSH-based similarity sharing reveals which samples resemble one another across institutions, a weaker guarantee than cryptographic methods, and one the authors themselves describe as "a heuristics model for privacy protection."

Differential privacy has been applied to gradient boosting in the centralised setting Li et al. [2020c], but these methods assume a single party with access to the full dataset and do not extend to federated or cross-device deployments.

Table 1 summarises the landscape. Every existing federated tree-learning method assumes that each participant holds many samples; none accommodates the regime in which each client contributes a single record. PrivateBoost occupies this gap: horizontal federated gradient boosting with one sample per client, information-theoretic security via Shamir secret sharing, and no requirement for client-to-client communication.

2.3 Secret Sharing and Additive Aggregation

Shamir secret sharing Shamir [1979b] distributes a secret value as evaluations of a random polynomial, such that any m shares suffice for reconstruction while fewer reveal nothing. Ben-Or, Goldwasser and Wigderson Ben-Or et al. [1988] showed that this primitive is sufficient for secure computation of arbitrary functions, and verifiable extensions Gennaro et al. [1998] allow honest parties to detect and recover from malicious behaviour.

The property that makes secret sharing particularly well suited to our setting is additive homomorphism: the sum of shares equals shares of the sum. As established in Section 4.1, gradient boosting requires only aggregate gradient sums per histogram bin to determine each

Table 1: Comparison of federated tree-learning approaches. HE = homomorphic encryption, SA = secure aggregation, DP = differential privacy, LSH = locality-sensitive hashing, SS = secret sharing. Privacy column includes the security level.

Method	**Privacy**	**Setting**	**Scale**	**Min. samples**
SecureBoost Cheng et al. [2021]	HE, computational	Vertical	Cross-silo	Cohort
Fed. Forest Liu et al. [2022]	Enc., computational	Vertical	Cross-silo	Cohort
SimFL Li et al. [2020b]	LSH, computational	Horizontal	Cross-silo	Cohort
Xie et al. Xie et al. [2022]	SS, information-theoretic	Vertical	Cross-silo	Cohort
FedTree Li et al. [2023]	HE/SA/DP, config.	Both	Cross-silo	Cohort
privateboost (ours)	Shamir SS, information-theoretic	Horizontal	Cross-device	1

split. This means that shareholders can sum the shares they receive without ever reconstructing individual contributions, and the aggregator recovers exactly the statistics it needs. The compatibility between what the model requires (additive aggregates) and what the primitive provides (additive homomorphism) is what allows PrivateBoost to operate in the single-sample regime without sacrificing model quality.

3 Problem Statement

We consider a set of c clients $\{P_i\}_{i=1}^{c}$, each holding a single labeled sample $(\mathbf{x}_i, y_i)$ where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and y_i is the corresponding label. This one-sample-per-client setting arises naturally in medical applications where individual patients contribute their own health records without institutional intermediaries.

The problem of privacy-preserving federated XGBoost in the cross-device setting can be stated as follows:

Given: c clients, each holding a single sample $(\mathbf{x}_i, y_i)$; a set of n shareholders; and a single aggregator.
Learn: An XGBoost ensemble of T trees that matches the accuracy of a centralized model trained on the union $\{(\mathbf{x}_i, y_i)\}_{i=1}^{c}$, while revealing only aggregate statistics to any party.
Security requirement: No party learns individual client data. Specifically:

- The aggregator observes only aggregate gradient sums and feature statistics, never individual values or client identifiers.
- Any coalition of fewer than m out of n shareholders learns nothing about individual client contributions (information-theoretic guarantee via Shamir secret sharing).
- No client-to-client communication is required, accommodating intermittently available mobile devices.

The collusion assumptions underlying these guarantees, including the required separation between shareholders and aggregator, are formalised in the threat model (Section 5.1). Residual collusion scenarios and their mitigations are discussed in Section 5.4.

3.1 Roles

We define three roles in the protocol:

Definition 1 (Client)**.** *A client P_i holds a single sample $(\mathbf{x}_i, y_i)$. In each protocol round, the client computes its contribution (feature statistics or gradient–Hessian pairs), encodes these as Shamir secret shares, and distributes shares to all n shareholders. Clients receive broadcast*

information (bin edges, split decisions) from the aggregator, but never send data to it directly. Clients do not communicate with each other.

Definition 2 (Shareholder)**.** *A shareholder S_j ($j = 1, \ldots, n$) receives secret shares from clients and computes partial sums over received shares. Shareholders forward these partial sums to the aggregator upon request. Individual shareholders observe only their own share of each client's contribution, which is information-theoretically independent of the secret value.*

Definition 3 (Aggregator)**.** *The aggregator A coordinates the training process. It collects partial sums from at least m shareholders, reconstructs aggregate statistics via Lagrange interpolation, determines bin edges and split decisions, and broadcasts these to clients. The aggregator never receives raw shares or client identifiers.*

3.2 Challenges

Adapting XGBoost to this setting introduces several challenges not present in standard federated or centralized training:

1. **No local computation.** Each client holds exactly one sample, so local gradient computation produces a single (g_i, h_i) pair per round rather than a batch. There are no local statistics to aggregate before communication—every individual value must be protected.

2. **Histogram construction without raw data.** XGBoost requires discretizing features into histogram bins across the full dataset. In our setting, no party observes more than one raw feature value, so bin edges must be derived from securely aggregated statistics.

3. **Intermittent availability.** Cross-device clients (e.g., patient smartphones) may go offline at any time. The protocol must tolerate client dropout without corrupting aggregated results or requiring restarts.

4. **Path privacy.** As a tree is built, each client's traversal path encodes information about its feature values (e.g., that a patient's glucose exceeds a split threshold). Shareholders who observe which nodes a client contributes to can infer this path, creating a side channel not present in centralized training.

Section 4.4 describes how our protocol addresses challenges (1)–(3), Section 4.5 addresses (4), and Section 5.1 formalizes the security guarantees.

4 Protocol

4.1 Background: Histogram-Based Split Finding

XGBoost Chen and Guestrin [2016] builds decision trees by greedily selecting splits that maximize a gain function. Each tree corrects the residual errors of the current ensemble: a split groups samples that share a similar prediction error, so that each resulting leaf can apply a targeted correction. Without discretization, the algorithm would need to evaluate every unique feature value as a candidate threshold. Histogram-based methods make this tractable by discretizing continuous features into B bins, reducing the search to B candidate thresholds per feature.

Each sample i contributes a gradient $g_i = \partial L / \partial \hat{y}_i$ and Hessian $h_i = \partial^2 L / \partial \hat{y}_i^2$ based on the current prediction $\hat{y}_i$ and true label y_i. Intuitively, the gradient captures the direction and magnitude of each sample's error—how much the current model needs to adjust for that sample. The Hessian captures the curvature of the loss, acting as a per-sample confidence

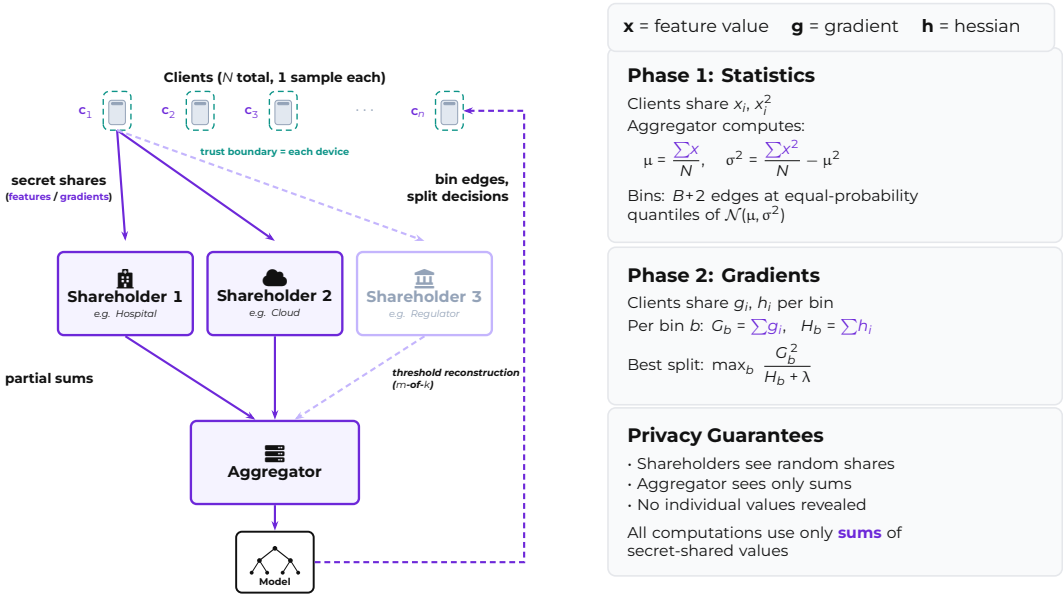

Figure 2: System architecture. Clients distribute Shamir shares to all shareholders. Shareholders sum received shares and forward partial sums to the aggregator, which reconstructs aggregate statistics (Σx, Σx^2 for binning) and gradient sums (ΣG, ΣH for splits). The aggregator broadcasts bin configurations and split decisions back to clients.

weight: where the loss surface is flat, the Hessian dampens the correction to prevent overfitting. For a candidate split that partitions samples into left (L) and right (R) sets, the gain is:

$$\text{Gain} = \frac{1}{2}\left[\frac{G_L^2}{H_L + \lambda} + \frac{G_R^2}{H_R + \lambda} - \frac{(G_L + G_R)^2}{H_L + H_R + \lambda}\right] \tag{1}$$

where $G_L = \sum_{i \in L} g_i$, $H_L = \sum_{i \in L} h_i$, and λ is a regularization parameter. The key insight is that split finding requires only *aggregate* gradient sums per bin, not individual gradients. Our protocol computes these sums without revealing individual contributions.

4.2 System Model

We consider c clients, each holding a single labeled sample $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is a feature vector and y_i is the label. A set of n *shareholders* (e.g., $n = 3$) act as intermediate aggregation points. A single *aggregator* coordinates the training process. Fig. 2 illustrates the overall system architecture.

The adversarial assumptions underlying this architecture are detailed in the threat model (Section 5.1).

4.3 Building Blocks

Shamir Secret Sharing. To share a secret value s, sample a random polynomial $f(x)$ of degree $m - 1$ with $f(0) = s$. Distribute shares $(i, f(i))$ for $i = 1, \ldots, n$ to n shareholders. Any m shares can reconstruct s via Lagrange interpolation; fewer than m shares reveal nothing about s.

Finite Field Arithmetic. Shamir sharing requires arithmetic over a finite field $\mathbb{F}_p$. Following Catrina and Saxena Catrina and Saxena [2010], we represent real-valued gradients and statistics as fixed-point integers: a value v is encoded as $\lfloor v \cdot 2^k \rceil \bmod p$, where k is the precision parameter. Negative values map to the upper half of the field ($p - |v \cdot 2^k|$). We choose $p = 2^{61} - 1$, a Mersenne prime that fits in a 64-bit integer and admits fast modular reduction via bit shifts.

With precision $k = 24$ ($\approx$7 decimal digits), this leaves 37 bits of headroom for accumulating sums from hundreds of clients without overflow.

Additive Homomorphism. For shares of values a and b, the sum of shares equals shares of $(a + b)$. This allows shareholders to sum their received shares without learning individual values, and the aggregator reconstructs only the sum.

Commitments. Each client generates a commitment $c = H(\text{round}\|\text{client_id}\|\text{nonce})$ using a fresh nonce per round. Shareholders only sum shares that have matching commitment hashes, ensuring consistent aggregation: if a client's share reaches some shareholders but not others, those shares are excluded rather than causing corrupted reconstruction. The aggregator sees only commitment hashes, not client identifiers.

4.4 Protocol Flow

The protocol has two phases: a one-time statistics phase to define histogram bins, followed by repeated gradient phases to build trees.

Phase 1: Statistics. Before training, the aggregator needs feature statistics to define histogram bin edges. Each client i creates Shamir shares of their feature values $x_{i,f}$ and squared values $x_{i,f}^2$ for each feature f, along with a commitment. Clients send shares to shareholders. The aggregator requests sums from shareholders, reconstructs $\sum x$ and $\sum x^2$, and derives per-feature mean μ and standard deviation σ.

From these statistics the aggregator places B bin edges at equal-probability quantiles of $\mathcal{N}(\mu, \sigma^2)$:

$$e_j = \mu + \sigma \cdot \Phi^{-1}\left(\frac{j+1}{B+2}\right), \quad j = 0, \ldots, B \tag{2}$$

where Φ^{-1} is the inverse normal CDF, computed via rational approximation. Together with two overflow bins at $\pm\infty$, this yields $B + 2$ bins whose boundaries adapt to each feature's location and scale. Because the statistics phase already collects the necessary aggregates, Gaussian quantile binning adds no communication cost compared to a simpler equal-width scheme. The aggregator broadcasts the resulting bin edges to all clients.

Phase 2: Gradient Rounds. For each tree and each depth level, the aggregator needs gradient sums per (feature, bin) combination. Each client i:

1. Computes their gradient g_i and Hessian h_i from current ensemble prediction
2. For each feature f, determines which bin b their value $x_{i,f}$ falls into
3. Creates Shamir shares of (g_i, h_i) with a fresh commitment
4. Sends shares to shareholders, tagged with $(f, b, \text{commitment})$

Shareholders accumulate shares grouped by (feature, bin, commitment). When the aggregator requests sums for a set of commitments, each shareholder returns summed shares per (feature, bin). The aggregator reconstructs gradient sums $G_{f,b} = \sum g_i$ and $H_{f,b} = \sum h_i$ for each bin, evaluates the gain formula for each possible split, selects the best split, and broadcasts the decision (feature, threshold) to clients. Clients update their node assignments, and the process repeats for the next depth level.

After reaching maximum depth, leaf values are computed from gradient sums and the tree is added to the ensemble. The entire gradient phase repeats for each tree.

Algorithm 1 summarizes the complete protocol.

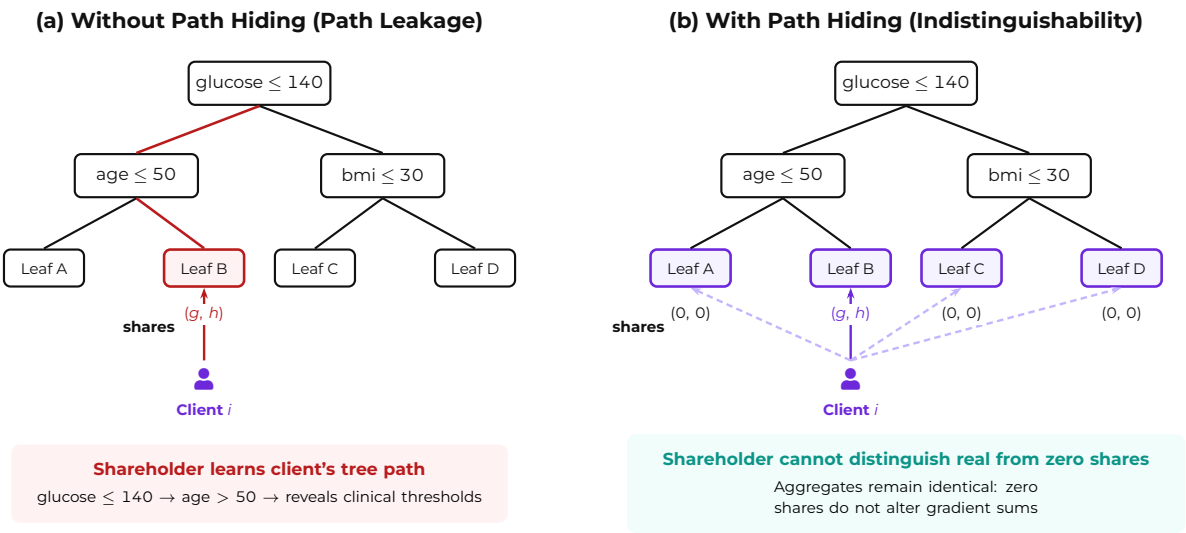

Figure 3: Path hiding mechanism. (a) Without path hiding, the client submits shares only to its true leaf node, revealing its tree path to shareholders. (b) With path hiding, the client submits shares to all leaf nodes: real gradient shares (g, h) to the true node and zero shares $(0, 0)$ to all others. Since Shamir shares of zero are indistinguishable from shares of other values, shareholders cannot identify the true node. The reconstructed aggregates are unaffected because zero shares do not alter gradient sums.

4.5 Path Hiding

In the basic protocol, each client submits gradient shares only for its current tree node, revealing the client's tree path to shareholders. For medical data, this can be sensitive—e.g., a shareholder learns that a patient's glucose exceeds a diagnostic threshold.

Path hiding eliminates this leakage. Instead of submitting shares for only its true node, each client submits shares for *all* active leaf nodes at the current depth (Fig. 3). For the true node, the client sends Shamir shares of its actual gradient and Hessian. For all other nodes, the client sends Shamir shares of zero. Since Shamir shares of zero are indistinguishable from shares of any other value, shareholders cannot determine which node is the client's true node.

The reconstructed aggregates are unaffected: zero-valued shares do not alter the gradient sums, so model quality is identical with and without path hiding. The cost is communication overhead—at depth d, the client submits to 2^d nodes instead of one, increasing per-round communication proportionally (quantified in Section 6.4). This mechanism is captured in lines 16–25 of Algorithm 1. Disabling path hiding reduces to submitting shares only for v_i^*, eliminating the loop over $\mathcal{V}_\ell$.

5 Security Analysis

5.1 Threat Model

We assume an honest-but-curious (semi-honest) adversary model: all parties follow the protocol correctly but may attempt to learn additional information from their observations.

Shareholders. At most $m - 1$ out of n shareholders may collude for an m-of-n threshold scheme. The collusion threshold is configurable: increasing from 2-of-3 to 3-of-5 or higher provides stronger guarantees while maintaining dropout tolerance. Shareholders enforce a minimum-client threshold: they refuse to release aggregate sums if fewer than a configurable minimum number of clients contributed shares, preventing reconstruction from insufficiently

anonymized aggregates. Crucially, shareholders do not collude with the aggregator; any such collusion would undermine the separation between share-level and aggregate-level views that the protocol relies on.

Aggregator. A single aggregator coordinates the protocol. It follows the protocol honestly but may attempt to learn individual values from the aggregates it reconstructs. It does not have access to raw shares or client identifiers.

Clients. Clients are assumed honest-but-curious. The protocol does not include cryptographic verification of individual contributions; defenses against malicious clients are discussed in Section 5.4.

5.2 Formal Security Proof

We prove security using the simulation paradigm Lindell [2017]. A protocol is secure if every adversary's view in the real execution can be *simulated* from the ideal output alone, meaning the adversary learns nothing beyond the intended output.

5.2.1 Ideal Functionality

We assume arithmetic over $\mathbb{F}_p$ with $p > n$ so that the n shareholder evaluation points $1, \ldots, n$ are distinct in $\mathbb{F}_p$. Sufficient headroom to prevent overflow of aggregate sums is a correctness requirement addressed in Section 4.3 ($p = 2^{61} - 1$ with $k = 24$ bits of precision leaves 37 bits for accumulation).

We use the following notation throughout this section. Each client i contributes a gradient $g_i = \partial L / \partial \hat{y}_i$ and Hessian $h_i = \partial^2 L / \partial \hat{y}_i^2$ (as defined in Section 4.1). Let F denote the number of features and $\mathcal{V}_\ell$ the set of leaf nodes at depth ℓ of the current tree. The hash function H produces outputs of length λ bits.

We define the ideal functionality $\mathcal{F}_{boost}$ as a trusted party that:

1. Receives feature vectors $(\mathbf{x}_i, y_i)$ from each client $i \in [c]$.

2. **Statistics phase.** Computes aggregate statistics $\sum_i x_{i,f}$ and $\sum_i x_{i,f}^2$ for each feature f. Returns these aggregate sums and the number of participating clients c to the aggregator (from which it derives bin edges). Returns the set of participating client pseudonyms to shareholders. Broadcasts bin edges to all clients.

3. **Gradient rounds.** In each round r (one depth level), receives per-client gradients (g_i, h_i) and per-feature bin assignments $\{b_{i,f}\}_{f=1}^F$ from the set of participating clients C_r. The trusted party determines each client's true node $v_i^* \in \mathcal{V}_\ell$ from the client's features and the prior split decisions. If $|C_r| < k_{\min}$ (the minimum-client threshold), outputs $\perp$ and halts the round. Otherwise, for each leaf node $v \in \mathcal{V}_\ell$, computes aggregate sums $G_{v,f,b} = \sum_{i \in S_{v,f,b}} g_i$ and $H_{v,f,b} = \sum_{i \in S_{v,f,b}} h_i$, where $S_{v,f,b}$ is the set of clients with $v_i^* = v$ assigned to bin b of feature f. Returns $\{(G_{v,f,b}, H_{v,f,b}, |S_{v,f,b}|)\}_{v,f,b}$ and $|C_r|$ to the aggregator.

4. **Shareholder leakage.** In the statistics phase, returns only the set of participating client pseudonyms to shareholders (not feature values). In gradient rounds, returns per-client per-feature bin assignments $\{b_{i,f}\}_{f=1}^F$ to shareholders, indexed by each client's persistent pseudonymous identifier. Shareholders do not learn which tree node is each client's true node (proved in Case (b) below). Cross-round linkability via transport-layer identity is acknowledged in Section 5.4.

5. Returns bin edges and split decisions to clients.

Commitment hashes appearing in the real protocol are not part of the ideal output; they are simulatable artifacts (see Case (a) below).

5.2.2 Security Statement

Theorem 1. *Under the semi-honest adversary model of Section 5.1, Protocol privateboost securely computes $\mathcal{F}_{boost}$ against:*

(a) a corrupt aggregator,

(b) a coalition of up to $m-1$ out of n shareholders (non-colluding with the aggregator),

(c) any subset of corrupt clients.

In cases (a) and (b), the Shamir component is perfectly *indistinguishable (information-theoretic); commitment hashes are indistinguishable in the random oracle model. In case (c), the view is deterministic given the client's input and the protocol output.*

Remark 1. *The theorem treats each party class independently. Cross-party collusion involving clients is not a separate case because clients observe only public broadcasts (bin edges and split decisions), so colluding with either the aggregator or a shareholder provides no information beyond what that party already possesses. The critical collusion scenario, between shareholders and the aggregator, is excluded by the threat model and discussed as a limitation in Section 5.4.*

Proof. We construct a simulator $\mathcal{S}$ for each case and show that the simulated view is indistinguishable from the real view.

Case (a): Corrupt aggregator. The aggregator's view spans both protocol phases. In each phase, it receives from shareholders: (i) the set of commitment hashes $\{c_{i,r}\}$ of participating clients (relayed by shareholders for aggregation grouping), and (ii) summed shares $\tilde{s}_j = \sum_i \phi_i(j)$ from each responding shareholder $j \in T$ ($|T| = m$), where $\phi_i(x)$ is client i's independently sampled Shamir polynomial with $\phi_i(0) = s_i$ (the client's secret value). In the statistics phase, the secret values are $s_i = x_{i,f}$ and $x_{i,f}^2$ for each feature f; in gradient rounds, they are $s_i = g_i$ and h_i for each (node, feature, bin) bucket.

Both phases have identical structure: the aggregator reconstructs an aggregate sum and observes commitment hashes via shareholders. We construct a single simulator that applies to both phases and to each (node, feature, bin) bucket independently.

Simulator $\mathcal{S}_A$: Given ideal output (aggregate statistics $\sum_i x_{i,f}$, $\sum_i x_{i,f}^2$, and the number of participating clients c for the statistics phase; $\{(G_{v,f,b}, H_{v,f,b}, |S_{v,f,b}|)\}_{v,f,b}$ and $|C_r|$ for each gradient round), for each aggregate sum S and responding shareholder set T:

1. Sample a uniformly random polynomial $q(x)$ of degree at most $m-1$ over $\mathbb{F}_p$ subject to $q(0) = S$.

2. Output $\{q(j)\}_{j \in T}$ as the simulated shareholder responses.

For commitment hashes, sample the appropriate number of independent uniformly random bitstrings of length λ (the hash output length): c commitments for the statistics phase, $|C_r|$ for each gradient round r.

Indistinguishability: In the real execution, $\tilde{s}_j = \sum_i \phi_i(j)$. Since each ϕ_i is an independently sampled uniformly random polynomial of degree at most $m-1$ with $\phi_i(0) = s_i$, the sum $\tilde{\phi}(x) = \sum_i \phi_i(x)$ is a uniformly random polynomial of degree at most $m-1$ with $\tilde{\phi}(0) = \sum_i s_i = S$ (the independence of the ϕ_i ensures that the non-constant coefficients of $\tilde{\phi}$ are uniformly distributed). The simulated polynomial $q(x)$ has the identical distribution: uniformly random of degree at most $m-1$ conditioned on $q(0) = S$. Therefore the shareholder responses at any m-element subset $T \subseteq \{1, \ldots, n\}$ are identically distributed in both worlds. This argument applies independently to each aggregate sum: per-feature statistics in the statistics phase, and per-(node, feature, bin) gradient sums in gradient rounds.

Commitment hashes are computed as $H(\text{round}\|\text{id}\|\text{nonce})$ with a fresh random nonce per round. In the random oracle model, these outputs are uniformly distributed and independent,

matching the simulated random strings. The Shamir component of the view is therefore perfectly indistinguishable; the commitment component is indistinguishable in the random oracle model.

Case (b): Coalition of up to $m-1$ shareholders. Each corrupt shareholder j observes, for every client i and every round, shares covering all leaf nodes at the current depth. For each node $v \in \mathcal{V}_\ell$ and each feature f, the shareholder receives two shares $(j, \phi^g_{i,v,f}(j))$ and $(j, \phi^h_{i,v,f}(j))$, where $\phi^g_{i,v,f}$ and $\phi^h_{i,v,f}$ are independently sampled Shamir polynomials. Shares are tagged with the client's commitment hash, pseudonymous identifier, node v, and bin assignment $b_{i,f}$. At the true node v^*_i, the polynomials encode g_i and h_i respectively; at all other nodes they encode 0. In the statistics phase, shares are tagged with the client's pseudonym only.

Simulator $\mathcal{S}_B$: Given the ideal leakage (persistent pseudonymous identifiers and per-client per-feature bin assignments $\{b_{i,f}\}_{f=1}^F$ for gradient rounds, from $\mathcal{F}_{boost}$), for each client i, each corrupt shareholder j, each round r, each node $v \in \mathcal{V}_\ell$, and each feature f:

1. Sample $u^g_{i,j,r,v,f}, u^h_{i,j,r,v,f} \xleftarrow{\$} \mathbb{F}_p$ uniformly and independently.
2. Sample a commitment hash $c_{i,r}$ as a uniformly random bitstring of length λ (shared across all nodes and features for the same client and round).
3. Output $(j, u^g_{i,j,r,v,f})$ and $(j, u^h_{i,j,r,v,f})$ tagged with client i's persistent pseudonym, the commitment hash $c_{i,r}$, node v, and bin assignment $b_{i,f}$.

The simulator assigns consistent pseudonymous identifiers to each client across rounds, matching the cross-round linkability that shareholders observe in the real protocol.

Indistinguishability: Each Shamir polynomial $\phi^g_{i,v,f}$ and $\phi^h_{i,v,f}$ is sampled independently per (client, node, feature) tuple. By the privacy property of Shamir secret sharing, any $m-1$ evaluations of a uniformly random polynomial of degree at most $m-1$ are independently and uniformly distributed over $\mathbb{F}_p$, regardless of the free term Shamir [1979b]. This holds whether the free term is g_i (true node) or 0 (decoy node), and likewise for h_i. Therefore, a coalition of fewer than m shareholders cannot distinguish the true node from any decoy, and the real shares are i.i.d. uniform over $\mathbb{F}_p$, exactly matching the simulated values. Commitment hashes are uniformly random in the random oracle model (same argument as Case (a)). The Shamir component is therefore perfectly indistinguishable; the commitment component is indistinguishable in the random oracle model.

Case (c): Corrupt clients. Each client i observes only: (i) their own input $(\mathbf{x}_i, y_i)$, (ii) broadcast bin edges (derived from aggregate statistics), and (iii) broadcast split decisions per round (derived from aggregate gradient sums).

Simulator $\mathcal{S}_C$: Given client i's input and the ideal output (bin edges and split decisions), output these values directly.

Indistinguishability: The client's view is a deterministic function of its own input and the aggregator's broadcast messages, which are determined by the ideal output. There is nothing to simulate beyond what the ideal functionality provides.

Composition across rounds. The protocol comprises one statistics round and $T \cdot d$ gradient rounds (one per depth level per tree). The outputs of round r (split decisions) determine the inputs to round $r+1$ (bin assignments), so rounds are sequentially dependent. However, each round uses independently sampled Shamir polynomials and fresh commitment nonces (with distinct random oracle inputs), and the same n shareholders participate across all rounds with independent randomness per round. By the modular sequential composition theorem for semi-honest protocols Goldreich [2004], security of each individual round as a sub-functionality implies security of the composed protocol over all $T \cdot d + 1$ rounds.

When clients drop out between rounds, the aggregator and shareholders observe a changing participant count, which is captured by $|C_r|$ and the per-round sets $S_{f,b}$ in the ideal functionality. The Shamir simulation arguments are unaffected by the client set size, as they hold for any number of contributing clients $c \geq 1$. □

5.3 Communication and Computation Cost

The practical viability of the protocol depends on how its costs scale with the number of clients, features, and security parameters. Let c denote the number of clients, n the number of shareholders, m the reconstruction threshold, F the number of features, B the number of bins, T the number of trees, and D the maximum tree depth. We summarise the asymptotic behaviour here; Section 6.4 evaluates concrete communication volumes on the experimental datasets.

The dominant communication cost is secret sharing: in each gradient round (one depth level of one tree), every client distributes Shamir shares of $2F$ gradient–Hessian values to all n shareholders, producing $O(Fcn)$ shares per round and $O(TDFcn)$ over an entire training run of T trees at maximum depth D. A one-time statistics phase adds a further $O(Fcn)$ shares for the per-feature means and variances used to derive bin edges. In the reverse direction, each shareholder returns $O(FB)$ partial sums per round to the aggregator, totalling $O(TDFBn)$, while the aggregator broadcasts only a single split decision per round, a negligible cost. When path hiding is enabled (Section 4.5), clients submit shares for all active nodes at each depth rather than only their true node, increasing total client-to-shareholder communication from $O(TDFcn)$ to $O(T \cdot 2^D \cdot Fcn)$.

Computation is likewise dominated by polynomial operations. Creating a Shamir share requires evaluating a degree-$(m-1)$ polynomial at n points, costing $O(nm)$ per secret value; each client therefore performs $O(TDFnm)$ work over a full run. Shareholders sum $O(c)$ incoming shares per histogram bucket, and the aggregator reconstructs each of the $O(FB)$ aggregate values per round via Lagrange interpolation over m points, contributing $O(TDFBm^2)$ in total. Split finding adds a further $O(TDFB)$, which is subsumed by the reconstruction cost.

A useful property of this cost structure is that communication and client-side computation grow linearly in n, while reconstruction cost depends only on the threshold m. Increasing n while holding m fixed therefore improves fault tolerance, allowing more shareholders to be unavailable without disrupting the protocol, at no additional reconstruction cost. Conversely, raising m strengthens the collusion threshold at the price of higher-degree polynomials for sharing and $O(m^2)$ Lagrange reconstruction, a trade-off whose practical impact is evaluated in Section 6.4.

5.4 Limitations

The security proof establishes that no party learns more than the aggregate statistics it is entitled to, but several residual concerns remain outside the scope of the formal guarantee.

The most direct form of information leakage is structural: the aggregator necessarily learns the number of clients routed to each tree branch, and for branches with very few clients this narrows the anonymity set enough that auxiliary information could enable inference attacks. Adding calibrated noise via differential privacy (Section 7.1) would mitigate this by obscuring exact counts. A subtler channel arises from cross-round linkability: while commitment hashes are freshly generated each round, preventing the aggregator from linking submissions across rounds, shareholders can link a client's submissions via transport-layer identity such as a network address. This is not a protocol-level vulnerability but a transport-level one, straightforwardly mitigated by routing client submissions through an anonymous communication channel.

The protocol's trust assumptions also warrant scrutiny. The honest-but-curious model does not protect against adversarial behaviour by any party. A malicious client could submit poisoned gradients or out-of-range values to corrupt the model; in the extreme case, a Sybil attack could flood the protocol with fabricated identities. Two complementary defences apply: platform attestation mechanisms (such as App Check on Android and App Attest on iOS) verify that participating clients run legitimate, unmodified application code, while feature-specific clipping at the aggregation layer bounds the influence of any single contribution regardless of its origin. A malicious aggregator, conversely, could corrupt split decisions or model outputs;

this can be mitigated by running multiple independent aggregators that cross-verify results, or by applying verifiable computation techniques. The security proof further assumes that shareholders and the aggregator do not collude. If a shareholder forwards per-client shares to the aggregator, the effective collusion threshold is reduced: an aggregator colluding with k shareholders needs only $m-k$ additional compromised shareholders to reconstruct individual secrets. This risk can be managed by increasing the reconstruction threshold (e.g., from 2-of-3 to 3-of-5, with communication overhead evaluated in Section 6.4), by operating shareholders and the aggregator under independent organisations, or by running shareholder logic inside trusted execution environments whose remote attestation converts the non-collusion assumption into a hardware-enforced guarantee.

Finally, the protocol reveals exact aggregate sums to the aggregator, offering no formal differential privacy guarantee. However, the mean and variance statistics already computed during the binning phase provide a natural calibration point: Gaussian noise scaled to these statistics can be added to gradient sums, yielding formal (ε, δ)-differential privacy with no additional communication cost (Section 7.1).

6 Experiments

The clinical relevance of PrivateBoost depends on whether privacy-preserving aggregation from single-record clients can match the predictive quality of centralised training on the structured diagnostic data that predominates in medical practice. We evaluate this across four medical classification tasks of increasing scale. Three small datasets from the UCI Machine Learning Repository Dua and Graff [2019] represent common diagnostic scenarios: Heart Disease Detrano et al. [1989] (297 samples, 13 features), Breast Cancer Street et al. [1993] (569 samples, 30 features), and Pima Diabetes Smith et al. [1988] (768 samples, 8 features). To demonstrate scalability to cross-device settings, we additionally evaluate on the CDC Diabetes Health Indicators dataset from the Behavioral Risk Factor Surveillance System (BRFSS) Centers for Disease Control and Prevention [2015], a balanced subset of 70,692 samples with 21 health indicator features predicting diabetes diagnosis. In all experiments, each client holds exactly one record, the regime in which a patient contributes a single diagnostic measurement from a personal device. We report results over 5 stratified random 80/20 train/test splits (mean $\pm$ std).

Following the protocol design, each client holds exactly one sample: we instantiate one client per training example. We use a 2-of-3 Shamir threshold, 10 bins per feature, max depth 3, 15 trees, learning rate 0.15, regularization $\lambda = 2.0$, and logistic loss. Bin edges are placed at Gaussian quantile boundaries as described in Section 4.4. We compare against two XGBoost baselines: (1) XGBoost with matched tree hyperparameters (depth, trees, learning rate, λ, and 10 bins), and (2) XGBoost with default settings.

6.1 Model Quality

The central question is whether the cryptographic overhead of secret sharing exacts a cost in predictive accuracy. Fig. 4 shows that it does not: PrivateBoost converges at a similar rate to centralised XGBoost across all four datasets, typically reaching its final AUC-ROC within 8–10 trees. The learning curves overlap or closely track the matched XGBoost configuration, confirming that privacy-preserving aggregation does not impede the gradient boosting optimisation.

Table 2 reports final classification metrics. On Heart Disease and CDC BRFSS, PrivateBoost matches XGBoost with identical hyperparameters exactly (81.7% and 74.2% accuracy, 0.880 and 0.818 AUC-ROC, respectively). On Breast Cancer and Pima Diabetes, differences fall within one standard deviation. The largest dataset, CDC BRFSS with 70,692 individual clients, is the most telling: PrivateBoost achieves 0.818 AUC-ROC, identical to centralised XGBoost,

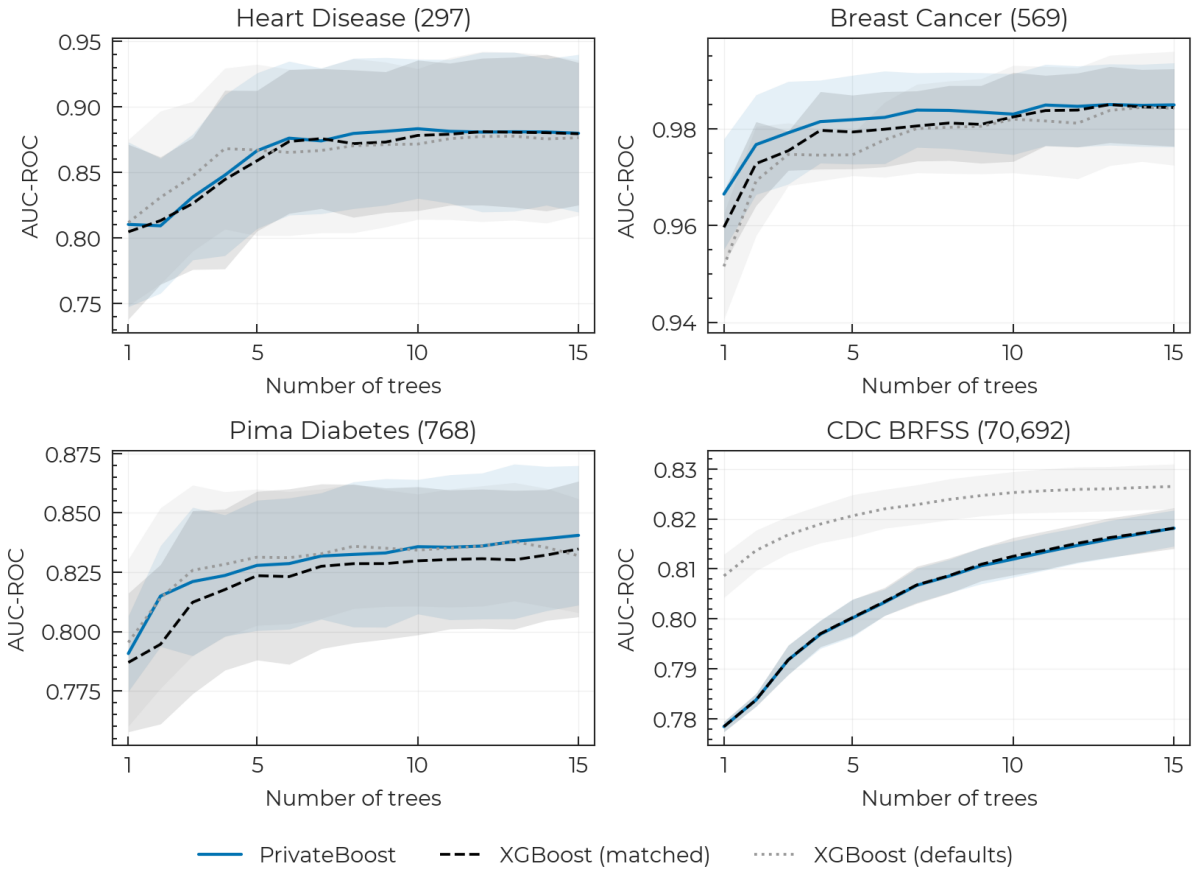

Figure 4: AUC-ROC vs. number of trees for *privateboost*, XGBoost with matched hyperparameters, and XGBoost with defaults across four medical datasets. Shaded bands show ±1 standard deviation over 5 stratified random splits.

demonstrating that the protocol introduces no measurable utility loss even when every client contributes a single record.

6.2 Binning Analysis

Because PrivateBoost cannot access raw feature values, it must approximate optimal split thresholds through histogram binning derived solely from aggregate mean and variance. The quality of this approximation depends on how bin edges are placed. We compare two strategies: Gaussian quantile binning, which places edges at equal-probability quantiles of $\mathcal{N}(\mu, \sigma^2)$, and equal-width binning, which spaces edges at equal intervals across the same range.

Table 4 quantifies bin uniformity using the coefficient of variation (CV) of bin occupancy, where CV $= 0$ corresponds to perfectly uniform bins. Gaussian quantile binning produces significantly more uniform bins on continuous features (mean CV 0.59 vs. 0.66, Wilcoxon signed-rank $p = 9.6 \times 10^{-8}$), while discrete features are identical under both methods. However, Table 3 shows that this uniformity advantage does not translate to better split quality. We measure this via gain retention: the ratio $r = G_{\text{binned}}/G_{\text{exact}}$ of the binned split gain to the exact optimum obtained by iterating over all unique feature values. Both methods retain approximately 87% of the exact gain on continuous features with no statistically significant difference (Mann–Whitney $p = 0.80$). Discrete features achieve near-perfect retention regardless of method, since the few distinct values map cleanly to bins. This indicates that bin uniformity alone does not determine split quality. That both privacy-compatible strategies recover approximately 87% of the exact per-split gain, yet produce identical AUC-ROC to centralised XGBoost (Section 6.1), suggests

Table 2: Classification metrics (mean ± std over 5 splits). PB = PrivateBoost, XGB-M = XGBoost matched hyperparameters, XGB-D = XGBoost defaults.

Dataset	**Method**	**Accuracy**	**AUC-ROC**	**F1**
Heart Disease	PB	81.7 ± 6.7	$.880 \pm .060$	$.792 \pm .073$
(297)	XGB-M	81.7 ± 5.0	$.879 \pm .054$	$.794 \pm .053$
	XGB-D	80.3 ± 4.9	$.876 \pm .059$	$.782 \pm .044$
Breast Cancer	PB	97.2 ± 0.7	$.985 \pm .009$	$.978 \pm .006$
(569)	XGB-M	96.1 ± 1.0	$.984 \pm .008$	$.970 \pm .008$
	XGB-D	95.2 ± 1.3	$.984 \pm .012$	$.962 \pm .011$
Pima Diabetes	PB	77.3 ± 1.9	$.841 \pm .029$	$.634 \pm .051$
(768)	XGB-M	76.5 ± 2.7	$.835 \pm .029$	$.622 \pm .064$
	XGB-D	76.0 ± 2.0	$.832 \pm .024$	$.648 \pm .013$
CDC BRFSS	PB	74.2 ± 0.3	$.818 \pm .004$	$.754 \pm .003$
(70,692)	XGB-M	74.2 ± 0.3	$.818 \pm .004$	$.754 \pm .003$
	XGB-D	74.9 ± 0.4	$.826 \pm .004$	$.762 \pm .003$

Table 3: Gain retention ($G_{\text{binned}}/G_{\text{exact}}$, higher is better) pooled across all four datasets (5 random splits, 15 trees, depth 3, $B = 10$ bins), stratified by feature type. Continuous: > 20 unique values; discrete: ≤ 20. The p-value is from a two-sided Mann–Whitney U test.

Features	Gaussian	Equal-width	p
Cont. (45)	0.871	0.877	0.801
Disc. (27)	0.997	0.999	0.253
All (72)	0.913	0.920	0.139

that the ensemble structure of gradient boosting is robust to modest per-split approximation error.

Fig. 5 illustrates the mechanisms on two continuous Pima Diabetes features. On near-symmetric Glucose ($\gamma_1 = 0.17$), Gaussian quantile binning distributes bin edges more evenly across the data-dense region. On the heavily skewed Insulin feature ($\gamma_1 = 2.27$), both methods concentrate approximately 49% of samples in a single bin due to a zero-inflated spike, limiting the resolution of either approach.

6.3 Client Dropout Resilience

In a deployment where patients contribute from personal devices, intermittent connectivity and unpredictable participation are the norm rather than the exception. We evaluate robustness to this by randomly excluding a fraction of clients each gradient round, with independent sampling per round. Dropout is applied only to gradient rounds; all clients participate in the one-time statistics phase that defines bin edges. In domains with well-established features such as standard clinical measurements, bin edges can alternatively be derived from known population statistics without requiring client participation at all.

As shown in Fig. 6, AUC-ROC remains stable across all datasets even at 80–90% dropout, with only a modest increase in variance. This resilience is structural: Shamir reconstruction depends on shareholder availability, not on the number of contributing clients, and sharehold-

Table 4: Bin occupancy uniformity (mean CV $= \sigma/\mu$ of bin counts, lower is better) pooled across all four datasets, stratified by feature type. The p-value is from a two-sided Wilcoxon signed-rank test (paired by feature).

Features	Gaussian	Equal-width	p
Cont. (45)	0.59	0.66	9.6×10^{-8}
Disc. (27)	2.24	2.24	—
All (72)	1.20	1.25	9.6×10^{-8}

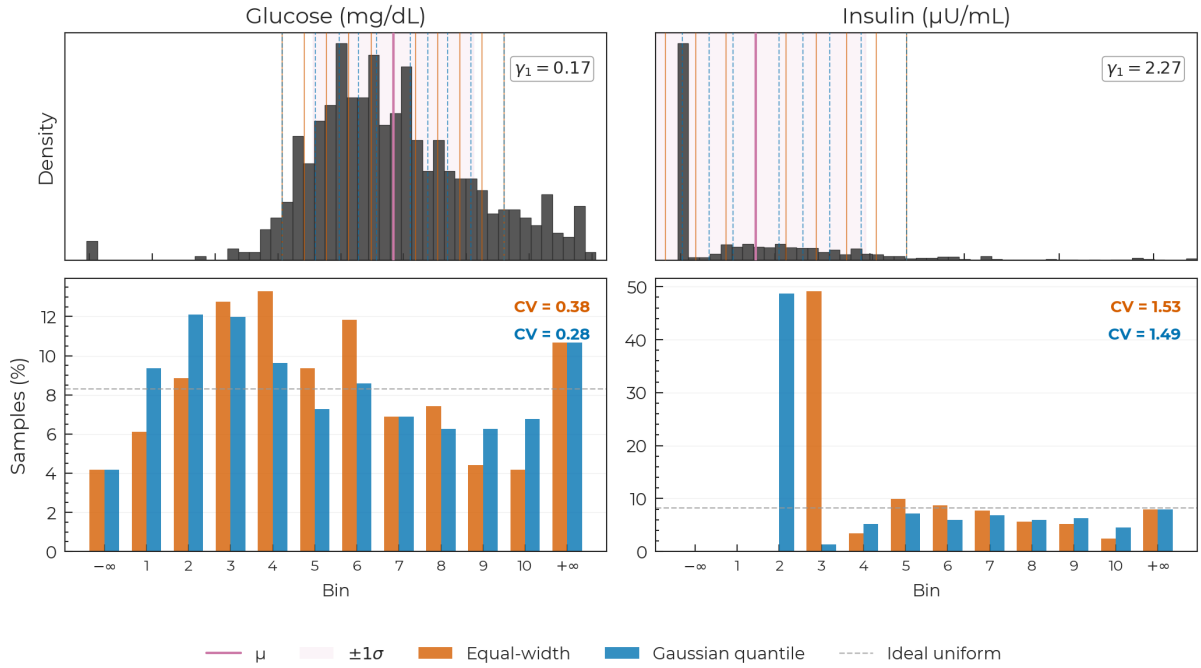

Figure 5: Bin occupancy comparison on two Pima Diabetes features with contrasting skewness (γ_1). Top row: feature distributions with equal-width (amber, solid) and Gaussian quantile (blue, dashed) bin edges overlaid. Bottom row: per-bin sample fractions under both methods (grouped bars); dashed gray line indicates ideal uniform occupancy. The coefficient of variation (CV) of bin occupancy is annotated for each method.

ers sum whichever shares they receive. Moreover, the threshold scheme means that not all shareholders need to be available at any given time; reconstruction requires only m of n shareholder responses, providing fault tolerance at both the client and infrastructure level. The minimum-client threshold enforced by shareholders (Section 5.1) provides a lower bound, preventing reconstruction from insufficiently anonymised aggregates.

6.4 Communication Cost

For a protocol targeting personal devices, communication cost must remain within the constraints of a standard mobile data plan. We measure this by computing the serialised size of each protocol message as it would appear on the wire. Each gradient share consists of a commitment (32 bytes), routing fields (round identifier, depth, node index; 24 bytes), and the gradient histogram ($2 \times F \times B_{\text{total}} \times 8$ bytes, where $B_{\text{total}} = B + 2$ includes overflow bins). Each client sends one share per shareholder per gradient round. The totals include all three traffic directions: client-to-shareholder, shareholder-to-aggregator, and aggregator broadcast (bin

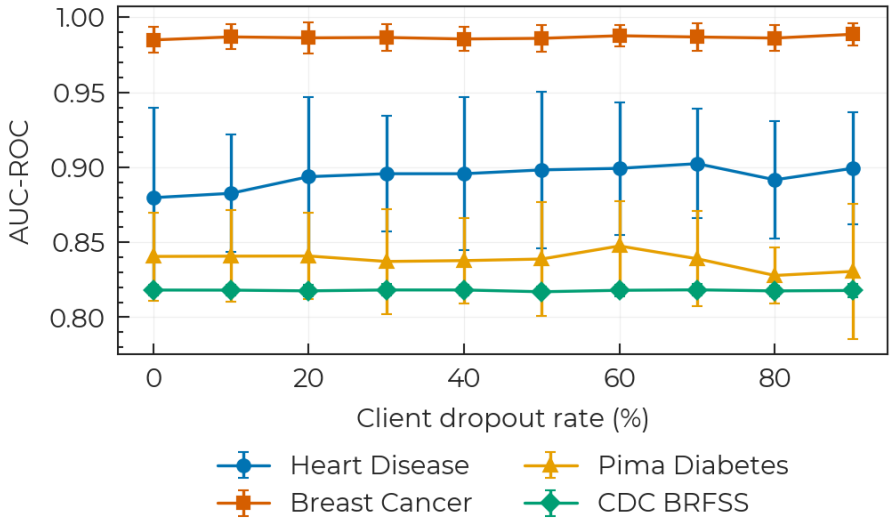

Figure 6: AUC-ROC under varying client dropout rates.

Table 5: Per-client communication cost (KB) under different Shamir thresholds and path hiding. 15 trees, max depth 3.

	2-of-3		**3-of-5**	
Dataset	**Base**	**+Hiding**	**Base**	**+Hiding**
Heart Disease (297)	379	867	624	1,438
Breast Cancer (569)	847	1,957	1,405	3,254
Pima Diabetes (768)	239	544	392	901
CDC BRFSS (70,692)	597	1,378	988	2,289

configurations and split decisions).

Table 5 reports per-client communication costs across all four datasets under two Shamir thresholds. With a 2-of-3 threshold, per-client cost ranges from 239 KB (Pima Diabetes, 8 features) to 847 KB (Breast Cancer, 30 features), scaling linearly with the number of features. Increasing to a 3-of-5 threshold, which provides stronger collusion resistance, increases cost by approximately 1.65×. On CDC BRFSS, each client transmits 597 KB with 2-of-3 or 988 KB with 3-of-5; for comparison, a single smartphone photograph is typically several megabytes, placing the entire training contribution of a patient well within routine mobile usage.

Path hiding (Section 4.5) increases per-client cost by 2.33× at depth 3, consistent with the $O(T \cdot 2^D \cdot Fcn)$ scaling derived in Section 5.3. Fig. 7 shows the resulting trade-off between model quality and communication cost as a function of tree depth. AUC-ROC improves rapidly from depth 1 to depth 3, then plateaus on all datasets, while communication cost continues to grow and the path hiding overhead grows exponentially with depth. This pattern is consistent across datasets regardless of size, motivating our default of depth 3: it captures most of the model quality while keeping path hiding overhead manageable.

The observed overhead is significantly lower than the worst-case $2^D = 8\times$ bound because depth 0 always has exactly one active node (no overhead), and most gradient boosting trees are shallow ($D \leq 3$), limiting the exponential growth. Crucially, path hiding does not affect model quality, since zero-valued shares for non-true nodes do not alter the reconstructed aggregates.

7 Future Work

The current protocol provides information-theoretic privacy for honest-but-curious participants. Several extensions would strengthen its guarantees and broaden its applicability.

7.1 Differential Privacy

Adding calibrated Gaussian noise to gradient sums before reconstruction would provide formal (ε, δ)-differential privacy Dwork and Roth [2014], a guarantee that has been successfully applied to deep learning Abadi et al. [2016]. The mean and variance statistics already computed during the binning phase provide a natural calibration point for the noise scale, requiring no additional communication. Per-tree privacy budgets with composition accounting across the ensemble would allow practitioners to tune the privacy–utility trade-off for their specific clinical application.

7.2 Input Validation and Feature Clipping

The honest-but-curious model assumes that clients submit truthful values, but a malicious or compromised device could submit arbitrarily large gradients to disproportionately influence the model. Because the aggregator already knows the per-feature mean and variance from the statistics phase, feature-specific clipping bounds can be derived and broadcast to clients, who enforce them locally before creating shares. This would limit the influence of any single contribution, providing a practical defence against gradient poisoning that complements the platform attestation mechanisms discussed in Section 5.4.

8 Conclusion

We have presented PrivateBoost, a privacy-preserving federated gradient boosting framework for cross-device medical learning in the extreme non-IID regime where each participant contributes a single record. By combining Shamir secret sharing with a simple and robust three-party architecture, the system provides information-theoretic privacy guarantees without requiring client-to-client coordination or reliance on institutional intermediaries. Across four medical datasets, comprising up to 70,692 individual clients, PrivateBoost matches centralised XGBoost in predictive performance, while remaining resilient to high and unpredictable dropout rates and operating comfortably within the bandwidth constraints of personal devices. These results demonstrate that accurate and scalable learning is achievable even under severe data fragmentation, without compromising either privacy or practicality. More fundamentally, this work shows that the limitations of existing federated learning systems in patient-level settings are not intrinsic but arise from mismatches between model design and privacy constraints. By exploiting the compatibility between additive model structures and secure aggregation, PrivateBoost demonstrates that learning from single-record clients is both feasible and efficient. More broadly, this work enables a shift in how medical AI systems may be designed and deployed. Rather than aggregating data within institutions, learning can take place directly across patient-owned data, with individuals contributing to global models while retaining full control over their information. In this sense, PrivateBoost offers not merely a technical solution, but a step towards a more decentralised and patient-centric foundation for healthcare intelligence.

Acknowledgements

The authors would like to thank the Ministry of the Economy in Luxembourg and its digital health directorate for co-supporting this research. Similarly, we would like to thank the Luxembourg National Research Fund (FNR) for partially funding this research. This work was

supported in part by a PhD grant from the Luxembourg National Research Fund (FNR) under the project reference 17223919/MMS/Industrial Fellowship, and in part by the R-MMS clinical trial.

Declarations

- **Funding:** This work was supported in part by a PhD grant from the Luxembourg National Research Fund (FNR) under the project reference 17223919/MMS/Industrial Fellowship.

- **Conflict of interest:** The authors declare no competing interests.

- **Code availability:** The source code for *privateboost* is publicly available at `https://github.com/MyelinZ/privateboost`, released under the Apache 2.0 license.

References

Patrik Hummel, Matthias Braun, Max Tretter, et al. Data sovereignty: a concept for reconstructing data ethics and policy. *Health and Technology*, 11(1):17–22, 2021. doi: 10.1007/s12553-020-00496-y. URL `https://doi.org/10.1007/s12553-020-00496-y`.

Kevin Montgomery et al. A trust-based approach to consumer health data. *The Lancet Digital Health*, 3(1):e10, 2021. doi: 10.1016/S2589-7500(20)30283-3. URL `https://doi.org/10.1016/S2589-7500(20)30283-3`.

Christopher J Kelly, Alan Karthikesalingam, Suleyman Mustafa, Luke McLean, and Hugh Montgomery. Key challenges for delivering clinical impact with artificial intelligence. *Nature Medicine*, 25(10):1456–1464, 2019. doi: 10.1038/s41591-019-0584-6. URL `https://doi.org/10.1038/s41591-019-0584-6`.

Effy Vayena, Alessandro Blasimme, and I Glenn Cohen. Machine learning in medicine: Addressing ethical challenges. *PLOS Medicine*, 15(11):e1002689, 2018. doi: 10.1371/journal.pmed.1002689. URL `https://doi.org/10.1371/journal.pmed.1002689`.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Proceedings of the 20th International Conference on Artificial Intelligence and Statistics (PMLR)*, 54:1273–1282, 2017a. URL `https://proceedings.mlr.press/v54/mcmahan17a.html`.

Brendan McMahan, Ewan Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*, pages 1273–1282, 2017b. URL `https://proceedings.mlr.press/v54/mcmahan17a.html`.

Ittai Dayan, Holger R Roth, Abba Zhong, et al. Federated learning for predicting clinical outcomes in patients with covid-19. *Nature Medicine*, 27(10):1735–1743, 2021. doi: 10.1038/s41591-021-01506-3.

Nicola Rieke, Jonny Hancox, Wenqi Li, et al. The future of digital health with federated learning. *Nature Machine Intelligence*, 2(11):646–654, 2020. doi: 10.1038/s42256-020-00248-2.

Tian Li, Anit Kumar Sahu, Ameet Talwalkar, and Virginia Smith. Federated learning: Challenges, methods, and future directions. *IEEE Signal Processing Magazine*, 37(3):50–60, 2020a. doi: 10.1109/MSP.2020.2975205.

Ligeng Zhu, Zhijian Liu, and Song Han. Deep leakage from gradients. *Advances in Neural Information Processing Systems*, 32, 2019. URL https://proceedings.neurips.cc/paper/2019/hash/60a6c40c1e2e1840d1510853508007a4-Abstract.html.

Robert C Griggs, Mark Batshaw, Mary Dunkle, et al. Clinical research for rare disease: opportunities, challenges, and solutions. *Molecular Genetics and Metabolism*, 96(1):20–26, 2009. doi: 10.1016/j.ymgme.2008.10.003.

Amy Rogers, Gualberto De Paoli, Sonal Subherwal, et al. Decentralized clinical trials: the future of drug development? *Nature Reviews Drug Discovery*, 19(12):827–828, 2020. doi: 10.1038/d41573-020-00198-z. URL https://doi.org/10.1038/d41573-020-00198-z.

Léo Grinsztajn, Edouard Oyallon, and Gaël Varoquaux. Why do tree-based models still outperform deep learning on typical tabular data? *Advances in Neural Information Processing Systems*, 35:507–520, 2022. URL https://proceedings.neurips.cc/paper_files/paper/2022/hash/0378c6692b837c7a842fd25d345f850a-Abstract-Conference.html.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, et al. Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017a. doi: 10.1145/3133956.3133982.

Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979a. doi: 10.1145/359168.359176.

Keith Bonawitz, Hubert Eichner, Wolfgang Grieskamp, et al. Towards federated learning at scale: System design. *Proceedings of Machine Learning and Systems*, 1:374–388, 2019. URL https://proceedings.mlsys.org/paper/2019/hash/bd686fd640be98efaae0091fa301e613-Abstract.html.

Keith Bonawitz, Vladimir Ivanov, Ben Kreuter, Antonio Marcedone, H Brendan McMahan, Sarvar Patel, Daniel Ramage, Aaron Segal, and Karn Seth. Practical secure aggregation for privacy-preserving machine learning. *Proceedings of the 2017 ACM SIGSAC Conference on Computer and Communications Security*, pages 1175–1191, 2017b. doi: 10.1145/3133956.3133982.

James Henry Bell, Kallista A Bonawitz, Adrià Gascón, Tancrède Lepoint, and Mariana Raykova. Secure single-server aggregation with (poly) logarithmic overhead. *Proceedings of the 2020 ACM SIGSAC Conference on Computer and Communications Security*, pages 1253–1269, 2020. doi: 10.1145/3372297.3417885.

Kewei Cheng, Tao Fan, Yilun Jin, Yang Liu, Tianjian Chen, Dimitrios Papadopoulos, and Qiang Yang. Secureboost: A lossless federated learning framework. *IEEE Intelligent Systems*, 36 (6):87–98, 2021. doi: 10.1109/MIS.2021.3082561.

Yang Liu, Yingting Liu, Zhijie Liu, Junbo Zhang, Chuishi Meng, and Yu Zheng. Federated forest. *IEEE Transactions on Big Data*, 8(3):843–854, 2022. doi: 10.1109/TBDATA.2020.2992755.

Lunchen Xie, Jiaqi Liu, Songtao Lu, Tsung-Hui Chang, and Qingjiang Shi. An efficient learning framework for federated XGBoost using secret sharing and distributed optimization. *ACM Transactions on Intelligent Systems and Technology*, 13(5):1–28, 2022. doi: 10.1145/3523061.

Qinbin Li, Zhaomin Wu, Yanzheng Cai, Yuxuan Han, Ching Man Yung, Tianyuan Fu, and Bingsheng He. FedTree: A federated learning system for trees. *Proceedings of Machine Learning and Systems*, 5:89–103, 2023. URL https://proceedings.mlsys.org/paper_files/paper/2023/hash/3430e7055936cb8e26451ed49fce84a6-Paper-mlsys2023.pdf.

Qinbin Li, Zeyi Wen, and Bingsheng He. Practical federated gradient boosting decision trees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(04):4642–4649, 2020b. doi: 10.1609/aaai.v34i04.5895.

Qinbin Li, Zhaomin Wu, Zeyi Wen, and Bingsheng He. Privacy-preserving gradient boosting decision trees. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(01):784–791, 2020c. doi: 10.1609/aaai.v34i01.5422.

Adi Shamir. How to share a secret. *Communications of the ACM*, 22(11):612–613, 1979b. doi: 10.1145/359168.359176.

Michael Ben-Or, Shafi Goldwasser, and Avi Wigderson. Completeness theorems for non-cryptographic fault-tolerant distributed computation. *Proceedings of the Twentieth Annual ACM Symposium on Theory of Computing*, pages 1–10, 1988. doi: 10.1145/62212.62213.

Rosario Gennaro, Michael O Rabin, and Tal Rabin. Simplified vss and fast-track multiparty computations with applications to threshold cryptography. *Proceedings of the Seventeenth Annual ACM Symposium on Principles of Distributed Computing*, pages 101–111, 1998. doi: 10.1145/277697.277716.

Tianqi Chen and Carlos Guestrin. Xgboost: A scalable tree boosting system. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794, 2016. doi: 10.1145/2939672.2939785.

Octavian Catrina and Amitabh Saxena. Secure computation with fixed-point numbers. *Financial Cryptography and Data Security (FC)*, pages 35–50, 2010. doi: 10.1007/978-3-642-14577-3_6.

Yehuda Lindell. How to simulate it – a tutorial on the simulation proof technique. *Tutorials on the Foundations of Cryptography*, pages 166–239, 2017. doi: 10.1007/978-3-319-57048-8_6.

Oded Goldreich. *Foundations of Cryptography: Volume 2, Basic Applications*. Cambridge University Press, 2004. doi: 10.1017/CBO9780511721656.

Dheeru Dua and Casey Graff. UCI machine learning repository. `https://archive.ics.uci.edu/ml`, 2019.

Robert Detrano, Andras Janosi, Walter Steinbrunn, Matthias Pfisterer, Johann-Jakob Schmid, Sarbjit Sandhu, Kern H Guppy, Stella Lee, and Victor Froelicher. International application of a new probability algorithm for the diagnosis of coronary artery disease. *American Journal of Cardiology*, 64(5):304–310, 1989. doi: 10.1016/0002-9149(89)90524-9.

W Nick Street, William H Wolberg, and Olvi L Mangasarian. Nuclear feature extraction for breast tumor diagnosis. *IS&T/SPIE's Symposium on Electronic Imaging: Science and Technology*, 1905:861–870, 1993. doi: 10.1117/12.148698.

Jack W Smith, James E Everhart, WC Dickson, William C Knowler, and Robert S Johannes. Using the adap learning algorithm to forecast the onset of diabetes mellitus. *Proceedings of the Annual Symposium on Computer Application in Medical Care*, pages 261–265, 1988. URL `https://pmc.ncbi.nlm.nih.gov/articles/PMC2245318/`.

Centers for Disease Control and Prevention. Behavioral Risk Factor Surveillance System (BRFSS). `https://www.cdc.gov/brfss/`, 2015. Diabetes Health Indicators dataset, UCI Machine Learning Repository ID 891.

Cynthia Dwork and Aaron Roth. The algorithmic foundations of differential privacy. *Foundations and Trends in Theoretical Computer Science*, 9(3–4):211–407, 2014. doi: 10.1561/0400000042.

Martin Abadi, Andy Chu, Ian Goodfellow, H Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, pages 308–318, 2016. doi: 10.1145/2976749.2978318.

Algorithm 1 PrivateBoost Training (with Path Hiding)

Require: Clients $\{P_i\}_{i=1}^{c}$ with samples $(\mathbf{x}_i, y_i)$; shareholders $\{S_j\}_{j=1}^{n}$; aggregator A; threshold m; trees T; max depth D; bins B

Ensure: Trained XGBoost ensemble $\mathcal{E}$

Phase 1: Statistics

for each client P_i **do** ▷ In parallel
  $c_i \leftarrow H(\texttt{round} \| \texttt{id}_i \| \texttt{nonce})$
  Create Shamir shares of $x_{i,f}$ and $x_{i,f}^2$ for all features f
  Send shares with c_i to each shareholder
end for
Shareholders: sum received shares per feature, grouped by commitment
A: collect from $\geq m$ shareholders, reconstruct $\sum x_f$, $\sum x_f^2$
A: compute bin edges via Gaussian quantiles, broadcast to clients

Phase 2: Tree Building

$\mathcal{E} \leftarrow \emptyset$
for $t = 1, \ldots, T$ **do**
  Initialize root node containing all clients
  for depth $\ell = 0, \ldots, D-1$ **do**
    Let $\mathcal{V}_\ell$ be the set of all leaf nodes at depth ℓ
    for each client P_i with true node $v_i^* \in \mathcal{V}_\ell$ **do** ▷ In parallel
      $g_i \leftarrow \partial L / \partial \hat{y}_i$; $h_i \leftarrow \partial^2 L / \partial \hat{y}_i^2$
      Generate fresh commitment c_i
      for each leaf node $v \in \mathcal{V}_\ell$ **do** ▷ Path hiding: cover all nodes
        for each feature f **do**
          $b \leftarrow \textsc{Bin}(x_{i,f})$
          if $v = v_i^*$ **then** ▷ Client's true node
            Create Shamir shares of (g_i, h_i)
          else ▷ Decoy: indistinguishable from real shares
            Create Shamir shares of $(0, 0)$
          end if
          Send shares tagged (v, f, b, c_i) to shareholders
        end for
      end for
    end for
    for each $v \in \mathcal{V}_\ell$ **do**
      Shareholders: sum shares per (v, f, b), by commitment
      A: reconstruct $G_{f,b}, H_{f,b}$ for all (f, b) at v
    end for
    A: select best split (f^*, b^*) maximizing Gain
    A: broadcast split; clients update node assignments
  end for
  A: compute leaf values; $\mathcal{E} \leftarrow \mathcal{E} \cup \{\text{tree}_t\}$
end for
return $\mathcal{E}$

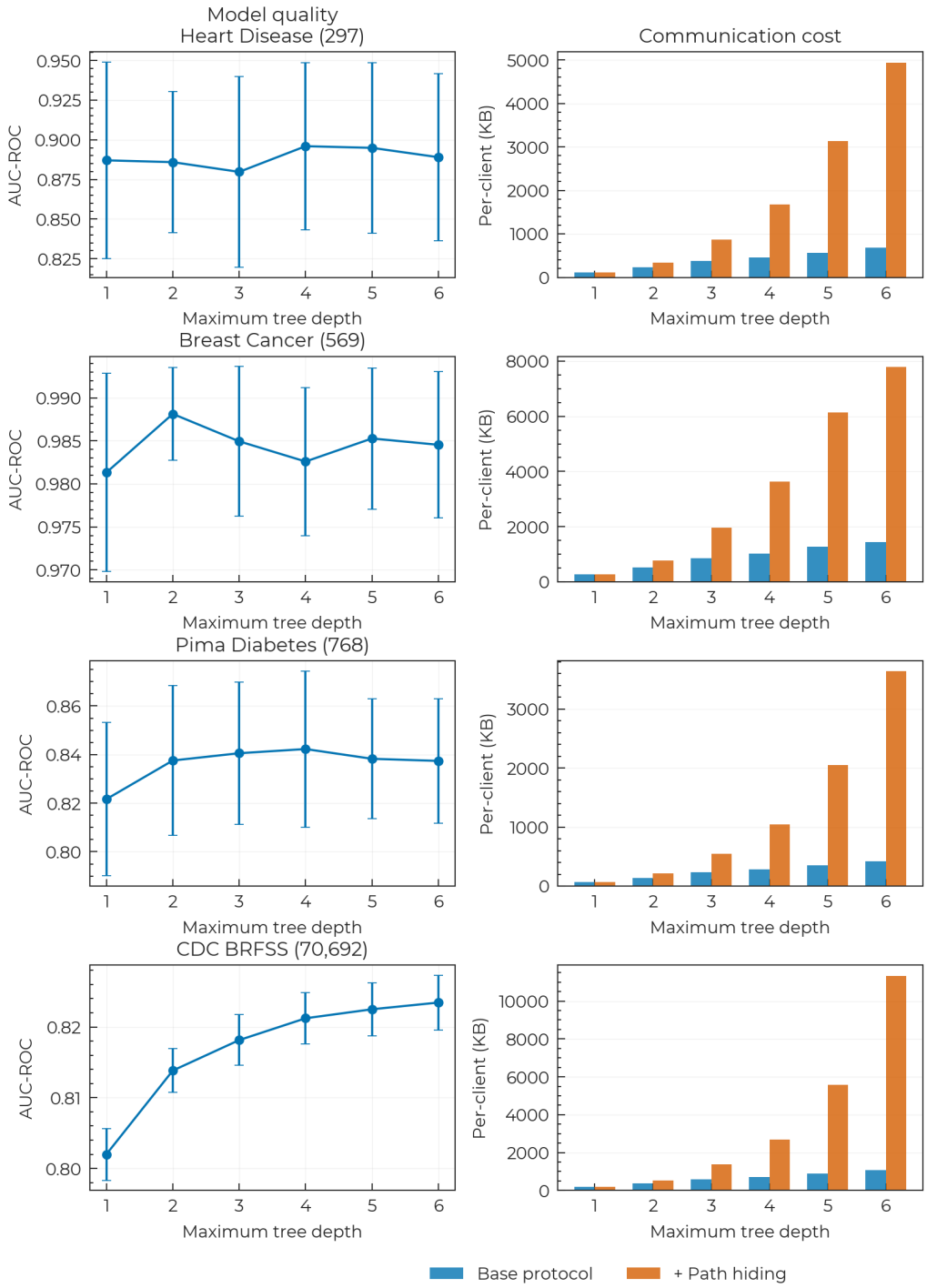

Figure 7: Impact of tree depth across all four datasets. Left column: AUC-ROC plateaus at depth 3 on all datasets. Right column: per-client communication cost (KB) with and without path hiding. Path hiding overhead grows exponentially with depth, while model quality saturates early.