

Design and Analysis of an Integrated AI-Driven and Blockchain-Enabled Connected Framework for Next-Generation Oncology Applications

Mian Usman Sattar

u.sattar@derby.ac.uk

University of Derby, United Kingdom.

Meznah A. Alamro

Princess Nourah bint Abdulrahman University

Alaeddine Mihoub

Qassim University

Soliman Aljarboa

Qassim University

Moez Krichen

FCSIT, Al-Baha University

Article

Keywords:

Posted Date: April 14th, 2026

DOI: <https://doi.org/10.21203/rs.3.rs-9288020/v1>

License:   This work is licensed under a Creative Commons Attribution 4.0 International License.

[Read Full License](#)

Additional Declarations: No competing interests reported.

Design and Analysis of an Integrated AI-Driven and Blockchain-Enabled Connected Framework for Next-Generation Oncology Applications

Mian Usman Sattar^{1,*}, Meznah A. Alamro^{2,*}, Alaeddine Mihoub³, Soliman Aljarboa^{3,*}, and Moez Krichen⁴

¹Department of Computing, University of Derby, Derby, DE22 1GB, UK

²Department of Information Technology, College of Computer and Information Sciences, Princess Nourah bint Abdulrahman University, P.O. Box 84428, Riyadh, 11671, Saudi Arabia

³Department of Management Information Systems, College of Business and Economics, Qassim University, Saudi Arabia

⁴FCSIT, Al-Baha University, Al-Baha 65528, Saudi Arabia

*meaalamro@pnu.edu.sa, u.sattar@derby.ac.uk, ssjrboa@qu.edu.sa

ABSTRACT

A wide variety of new cancer diagnostic methods have emerged alongside existing ones, though many still suffer from accuracy and accessibility issues. With the ever-increasing demand for timely and accurate cancer detection, the limitations need to be addressed to benefit patients. The research proposes a novel framework that leverages artificial intelligence (AI) and integrates it with blockchain technology to enhance oncology applications by addressing critical challenges in cancer diagnosis. As we become increasingly reliant on AI for medical diagnosis, data privacy, and transparency, a regulatory framework is becoming increasingly important. We propose a connected architecture based on federated learning that enables collaborative model training across institutions while keeping patient data decentralized and secure. By combining convolutional neural networks (CNNs) with blockchain for provenance tracking, we can achieve diagnostic accuracy, consistency, streamlined workflows, and increased stakeholder trust. A model optimised across diverse datasets without compromising patient privacy may accelerate the clinical application of AI-based diagnostic tools in oncology. This framework was validated using the PatchCamelyon and BreaKHis datasets in a non-IID federated setting. The EfficientNet-B0 baseline under centralized training achieved the AUC-ROC scores of 0.9557 and 1.0000 in the validation and test datasets, respectively. SimCLR-based fine-tuning yielded AUC-ROC scores of 0.9148 and 0.9928, which are highly competitive. In the federated setting, FedProx achieved the best performance ($\mu = 0.01$) with an AUC-ROC of 0.9241 on PCam and an F1-score of 0.9600 on BreaKHis. The integration of blockchain technology successfully maintained the model's utility with negligible loss. Thus, a federated oncology framework that can be secure, privacy-preserving, and auditable is certainly possible.

1 Introduction

The global AI healthcare sector is anticipated to grow to \$188 billion by 2030. This growth has accelerated the development of AI in medical diagnostics for treatment planning and cancer detection. Histopathological assessment is still regarded as the gold standard; however, sub-optimal manual analysis suffers from inter-observer variability and limited access to expert pathologists, resulting in disagreement in 25% of cases¹. The accuracy of AI-powered diagnostic systems has reached that of experts on benchmark datasets, such as PatchCamelyon and BreakHis^{2,3}. Nonetheless, their clinical application is prevented by data privacy regulations, high computational costs, and a lack of model interpretability⁴⁻⁶.

The use of AI, blockchain, and federated learning is very promising for oncology diagnostics. Histopathological images can be classified using deep learning models, particularly convolutional neural networks (CNNs). These fasten the diagnostic process and improve the detection of malignancy⁷. Yet, sharing data, which is crucial for effective AI training, becomes difficult due to compliance with privacy regulations such as HIPAA and GDPR⁵. In addition, generating extensive labeled datasets is expensive and time-consuming, and deep learning's 'black box' nature raises interpretability concerns⁸.

Present training deployment strategies, such as centralized training and transfer learning from natural images, are significantly challenged. A centralized approach can violate patient privacy, and transfer learning fails because medical images differ from natural images⁹. Researchers are studying federated learning (FL), which enables training

without sharing raw data, to build models^{4,10}. However, heterogeneous data across different institutions that are not independent and identically distributed (IID) make this FL challenging. Self-supervised learning techniques, such as contrastive learning, reduce reliance on labelled data^{11,12}.

This research aims to investigate how federated learning, self-supervised representation learning, and blockchain provenance can enhance model performance while preserving privacy and ensuring trust. By developing a comprehensive eight-module workflow for secure histopathology classification, we utilize federated learning for distributed training, SimCLR for self-supervised pretraining, and blockchain-inspired ledgers to track model provenance. Our methodology includes centralized baseline establishment, self-supervised pretraining, supervised fine-tuning, and federated learning alongside SHA-256 cryptographic fingerprinting and Grad-CAM-based explainability.

Our contributions are as follows:

1. A unified federated learning framework that integrates self-supervised pretraining (SimCLR), heterogeneity-aware aggregation (FedProx), and blockchain-enabled provenance tracking, validated on the PatchCamelyon and BreakHis datasets, achieving AUC-ROC scores of 0.9241 and 0.9855, respectively, under non-IID federated conditions while preserving data privacy and maintaining tamper-evident model auditability with negligible performance degradation upon blockchain integration.
2. An empirical evaluation demonstrating that SimCLR self-supervised pretraining provides a viable alternative to fully supervised initialization in data-scarce medical imaging scenarios, achieving AUC-ROC of 0.9148 on PCam and 0.9928 on BreakHis during centralized fine-tuning, and exhibiting consistent NT-Xent loss convergence over 10 pretraining epochs, thereby reducing dependence on large labeled datasets while retaining competitive discriminative performance relative to the fully supervised EfficientNet-B0 baseline.
3. A reproducible AI pipeline incorporating cryptographic model fingerprinting, append-only provenance ledgers, and Grad-CAM visual explanations, establishing transparency for regulatory compliance and clinical adoption by pathologists needing explainable decision support systems.

This paper is structured as follows: Section 2 provides the related work on the undertaken research topic. Section 3 details the study’s methodology that includes dataset preprocessing, baseline supervised training, SimCLR pretraining, non-IID client simulation, federated learning strategies, blockchain logging mechanisms, and explainability approaches applied to both the PatchCamelyon and BreakHis datasets. Section 4 presents extensive experimental results, including performance comparisons, ablation studies, convergence analyses, and clinical interpretability studies. Finally, Section 6 summarizes the main findings, discusses limitations, and outlines future work.

2 Related Work

This section provides an overview of key AI technologies integrated into privacy-preserving computational pathology and medical imaging systems.

2.1 Deep Learning in Histopathology

. The field of deep learning, particularly convolutional neural networks (CNNs), has reached the level of pathologists in tumor detection, grading, and subtype classification on whole-slide images, thus reducing pathologists’ workload while improving diagnostic consistency¹³. Tissue morphology and morphology-tissue treatment, such as segmentation of glands and nuclei detection of epithelial tumors, and direct genetic or prognostic markers prediction. The conventional clinical use of AI image recognition techniques based on deep learning is limited due to issues with stain and scanner variability, extreme resolution of images, data curation costs, and the necessity for further external validation and workflow integration^{14–16}.

2.2 Federated Learning for Medical Imaging

. The unique privacy and data-sharing constraints on patient information limit organizations’ ability to use large datasets to develop imaging AI models, thereby affecting sustainability. In federated learning (FL), the local model is trained on site-specific data. Only model updates are aggregated to provide near-centralized performance in classification and segmentation tasks while keeping patient data on-premise^{17,18}. Recent surveys and benchmarks certainly identify promise for cross-center medical imaging but also point to challenges arising from non-IID data, communication costs, and robustness against attacks or system heterogeneity¹⁹.

2.3 Self Supervised Representation Learning

One way to address the lack of expert-annotated images in medicine is to learn to discover useful image representations from unlabeled data via SSL. In the field of medical imaging, SSL consistently improves accuracy for downstream classification and segmentation tasks, increases label efficiency, and enhances out-of-distribution robustness compared with either fully supervised or ImageNet-only transfer learning. The performance is further enhanced in settings with strict data heterogeneity and few labels through transformer-based masked image modeling and hybrid semi-supervised schemes; thus, SSL naturally complements FL in a multi-center setting.

2.4 Blockchain-Based Provenance and Security

. Blockchain offers immutable, distributed ledgers for storing cryptographically linked records of data access, model updates, and clinical interactions, improving trust, traceability, and tamper-resistance in AI-enabled healthcare^{20, 21}. In the realm of IoT and e-health, the adoption of blockchains to preserve the integrity and audit of patients' medical records enables fine-grained access control and allows the orchestration of federated and AI workflows^{22, 23}. We are seeing more and more proposals to make provenance systems the backbone of regulated, multi-stakeholder AI. Approaches to explainable AI

The impactful decision-making in medical scenarios calls for explainable AI that can make the deep models used in imagology and histopathology transparent and clinically trustworthy²⁴. XAI techniques consist in saliency-based approaches, such as Grad-CAM or integrated gradients that highlight useful parts of images, and case based, text and auxiliary explanations, which include (only to name a few) prototype learning and uncertainty visualization²⁵. According to surveys, clinicians' trust in an AI system is supported by XAI, which is also regarded as useful for error analysis.

3 Proposed Approach

This research proposes a framework for federated learning that leverages blockchain provenance and explainable AI to develop a privacy-preserving mechanism for cancer detection. The strategy entails eight main Steps: compilation of the data and data preparation, baseline training in a supervised manner, pretraining in self-supervised mode by SimCLR, fine-tuning in the supervised mode, partitioning the data in a Non-IID manner, optimizing federated learning via FedAvg/FedProx, logging audit on Blockchain, and explainability by Grad-CAM. The architecture framework is illustrated in Fig. 1.

Algorithm 1 describes the complete eight-module privacy-preserving federated learning framework, including the cancer detection module. This implementation uses the PatchCamelyon (PCam) dataset, which contains 327,680 histopathological patches. The dataset is distributed across $K = 5$ clients over $T = 20$ communication rounds. Module 1 handles data collection and preprocessing. Raw PCam images of size 96×96 pixels are resized to 224×224 pixels to satisfy the input requirements of EfficientNet-B0, and standard data augmentation, including random horizontal and vertical flips, rotation, and color jittering, is applied to improve generalization. All images are normalized using ImageNet statistics and partitioned into training (262,144), validation (32,768), and test (32,768) splits. Module 2 establishes a centralized supervised baseline by training EfficientNet-B0 with ImageNet pretrained weights using binary cross-entropy loss and the Adam optimizer at a learning rate of 1×10^{-4} for 10 epochs, achieving a validation accuracy of 95.2%. This baseline serves as the upper bound for performance against which all federated models are evaluated. Modules 3 and 4 implement the self-supervised pretraining and fine-tuning pipeline, described in detail in Algorithm 2. The SimCLR framework pretrains the encoder f_ϕ on unlabeled data using contrastive learning for 50 epochs, after which the encoder is frozen and a linear classification head h_ψ is attached and trained for 10 epochs on labeled data. The resulting fine-tuned model serves as the global model initialization for all subsequent federated experiments, ensuring that federated training begins from a well-converged and meaningful feature space rather than random weight initialization. Module 5 partitions the dataset across $K = 5$ clients using a Dirichlet distribution with concentration parameter $\alpha = 0.8$ to simulate realistic Non-IID data heterogeneity across participating institutions. Module 6 executes federated training using either FedAvg or FedProx ($\mu \in \{0.01, 0.1\}$) for $T = 20$ communication rounds, as described in Algorithm 3, ensuring that no raw patient data leaves any local client environment at any time point during training. Module 7 links provenance logging based on the blockchain. It computes the SHA-256 cryptographic hash of every model update and appends it to an immutable ledger $\mathcal{L}$. Module 8 creates explainability visualizations using Grad-CAM that highlight the tissue regions most relevant for diagnosis, as shown in Algorithm 4.

The framework finishes with a real-time deployment process that accepts a new histopathological image x_{new} , preprocesses and normalizes the image, computes the score $p = \sigma(h_{\theta_g}(x_{tensor}))$, a sigmoid probability score, and classifies it as tumour or normal tissue using a fixed threshold $\tau = 0.5$. If the p value of a sample is greater than 0.8,

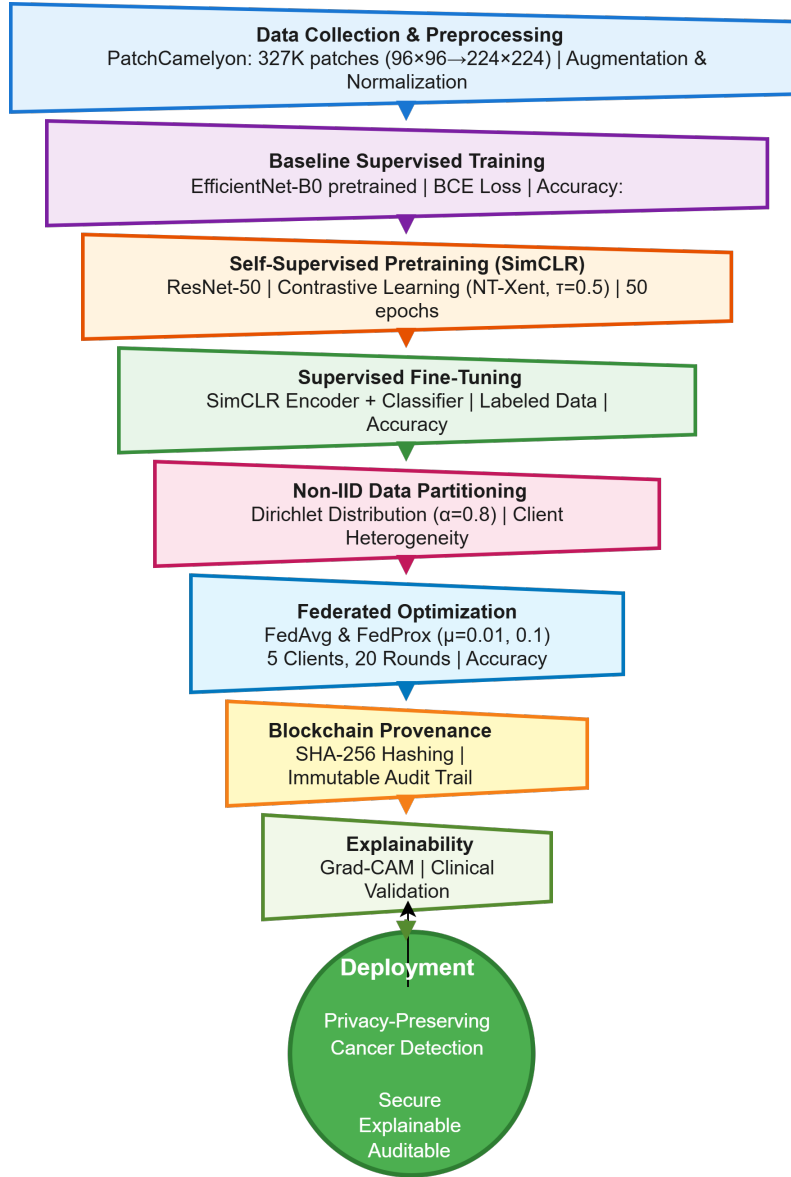


Figure 1. Federated Learning Framework for Privacy-Preserving Cancer Detection

it is flagged for urgent pathologist review. The generated structured clinical report will also include the prediction label, confidence level, Grad-CAM heatmap, and UTC timestamp for easy integration with EHR systems.

Algorithm 2 presents the SimCLR self-supervised pretraining procedure used in Module 3 of the framework. The algorithm accepts an unlabeled dataset $\mathcal{D}_u$, a batch size of $B = 256$, a temperature parameter $\tau = 0.5$, and trains for $E = 50$ epochs, outputting a pretrained encoder f_ϕ^* capable of extracting rich visual representations without requiring any labeled data. The network consists of an EfficientNet-B0 encoder $f_\phi : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1280}$ and a two-layer MLP projection head $g_\theta : \mathbb{R}^{1280} \rightarrow \mathbb{R}^{128}$, both optimized using the Adam optimizer at $\eta = 1 \times 10^{-3}$. For each image x_i in a mini-batch, the augmentation pipeline $\mathcal{T}$ applies random cropping, color jittering and Gaussian blur independently to generate two stochastically distinct augmented views $\tilde{x}_i^{(1)}$ and $\tilde{x}_i^{(2)}$, which constitute a positive pair. Both views are passed through the encoder and projection head to yield ℓ_2 -normalized embeddings $z_i^{(1)}$ and $z_i^{(2)}$ in the contrastive embedding space. The core learning objective is the normalized temperature-scaled cross-entropy (NT-Xent) loss, which attracts positive pairs in the embedding space while simultaneously pushing all other pairs, termed negative pairs, apart. For each sample i , the per-sample contrastive loss is (Eq. 1):

Algorithm 1: Privacy-Preserving Federated Learning Framework for Cancer Detection

Input: Dataset $\mathcal{D}$ (327,680 histopathology patches), clients $K = 5$, rounds $T = 20$, threshold $\tau = 0.5$
Output: Global model θ_g , blockchain ledger $\mathcal{L}$, explainability maps, clinical report

// Module 1: Data Preprocessing

- 1 Resize images $96 \times 96 \rightarrow 224 \times 224$; apply random flip, rotation, color jitter;
- 2 Normalize: mean=[0.485, 0.456, 0.406], std=[0.229, 0.224, 0.225];
- 3 Split: Train=262,144 | Val=32,768 | Test=32,768;

// Module 2: Baseline Supervised Training

- 4 Initialize EfficientNet-B0 with ImageNet weights;
- 5 Train with BCE loss, Adam ($lr = 1 \times 10^{-4}$), 10 epochs $\rightarrow$ Baseline accuracy: 95.2%;

// Module 3 & 4: SimCLR Pretraining + Fine-Tuning (Algorithm 2)

- 6 Pretrain SimCLR encoder f_ϕ on unlabeled data for 50 epochs ($\tau = 0.5$);
- 7 Freeze f_ϕ ; attach classifier head h_ψ ; fine-tune 10 epochs $\rightarrow$ Accuracy: 94.1%;

// Module 5 & 6: Non-IID Partitioning + Federated Training (Algorithm 3)

- 8 Partition $\mathcal{D}$ across $K = 5$ clients using Dirichlet($\alpha = 0.8$);
- 9 Run FedAvg / FedProx ($\mu \in \{0.01, 0.1\}$) for $T = 20$ rounds $\rightarrow$ Federated accuracy: 92.8%;

// Module 7: Blockchain Provenance (Algorithm 4)

- 10 Log each model update as SHA-256 hash to immutable ledger $\mathcal{L}$;

// Module 8: Explainability + Real-Time Deployment

- 11 Generate Grad-CAM heatmaps for clinical validation (Algorithm 4);

// Real-Time Inference on New Image x_{new}

- 12 Preprocess $x_{new} \rightarrow$ normalize $\rightarrow x_{tensor}$;
- 13 $p \leftarrow \sigma(h_{\theta_g}(x_{tensor}))$; **// Sigmoid probability score**
- 14 **if** $p > \tau$ **then**
- 15 | prediction $\leftarrow$ "Tumor Detected"; class_label $\leftarrow 1$;
- 16 **else**
- 17 | prediction $\leftarrow$ "Normal Tissue"; class_label $\leftarrow 0$;
- 18 $\mathcal{M}_{Grad-CAM} \leftarrow \text{GradCAM}(\theta_g, x_{tensor}, \text{class_label})$;
- 19 **if** $p > 0.8$ **and** class_label = 1 **then**
- 20 | **Alert:** High-confidence tumor $\rightarrow$ flag for urgent pathologist review;
- 21 Output clinical report: {prediction, $p \times 100\%$, $\mathcal{M}_{Grad-CAM}$, timestamp};
- 22 **return** θ_g , ledger $\mathcal{L}$, clinical report

$$\ell_i = -\log \frac{\exp(s_{i,i^+}/\tau)}{\sum_{j=1}^{2B} \mathbb{1}_{[j \neq i]} \exp(s_{i,j}/\tau)} \quad (1)$$

where $s_{i,j} = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}$ is the cosine similarity between embeddings, i^+ denotes the index of the positive pair (the second augmented view of the same image), and $\tau = 0.5$ sharpens the similarity distribution to improve contrastive discrimination. The batch-level loss $\mathcal{L}_{NT-Xent} = \frac{1}{2B} \sum_{i=1}^B (\ell_i + \ell_{i^+})$ treats both augmented views symmetrically as anchors. Encoder parameters ϕ and projection head parameters θ are updated via backpropagation at every iteration. Upon completion of training, the projection head g_θ is discarded as it serves only as an auxiliary component during contrastive optimization. The pretrained encoder f_ϕ^* is retained as a universal feature extractor and used for supervised fine-tuning and federated learning. This self-supervised approach is particularly well-suited to medical imaging, where labeled annotations are expensive and require significant clinical expertise. At the same time, large volumes of unlabeled histopathological images are routinely available from clinical workflows.

Algorithm 3 presents the unified Non-IID data partitioning and federated training procedure. The algorithm accepts the complete dataset $\mathcal{D}$, distributes it across $K = 5$ clients using a Dirichlet-based partitioning strategy, and then executes $T = 20$ rounds of federated optimization using either FedAvg (when $\mu = 0$) or FedProx (when $\mu \in \{0.01, 0.1\}$), outputting the trained global model θ_g and per-round performance metrics. Rather than distributing data uniformly across clients, the algorithm employs a Dirichlet distribution with concentration parameter $\alpha = 0.8$ to introduce realistic label heterogeneity. For a class $c \in \{0, 1\}$, we sample a proportion vector $\mathbf{p}_c \sim \text{Dirichlet}(\alpha, \dots, \alpha)$

Algorithm 2: SimCLR Self-Supervised Pretraining

Input: Unlabeled dataset $\mathcal{D}_u$, batch size $B = 256$, temperature $\tau = 0.5$, epochs $E = 50$

Output: Pretrained encoder f_ϕ^*

// Initialize

1 Initialize EfficientNet-B0 encoder $f_\phi : \mathbb{R}^{224 \times 224 \times 3} \rightarrow \mathbb{R}^{1280}$;

2 Initialize 2-layer MLP projection head $g_\theta : \mathbb{R}^{1280} \rightarrow \mathbb{R}^{128}$;

3 Initialize Adam optimizer, $\eta = 1 \times 10^{-3}$;

// Self-Supervised Training

4 **for** epoch $e = 1$ **to** E **do**

5 **for** each mini-batch $\mathcal{B} = \{x_i\}_{i=1}^B$ from $\mathcal{D}_u$ **do**

 // Generate two augmented views per image

6 $\tilde{x}_i^{(1)}, \tilde{x}_i^{(2)} \leftarrow \mathcal{T}(x_i), \mathcal{T}(x_i)$; // random crop, color jitter, Gaussian blur

 // Extract and project representations

7 $z_i^{(1)} \leftarrow g_\theta(f_\phi(\tilde{x}_i^{(1)}))$, $z_i^{(2)} \leftarrow g_\theta(f_\phi(\tilde{x}_i^{(2)}))$; // ℓ_2 -normalized

 // Compute NT-Xent contrastive loss

8 $\ell_i = -\log \frac{\exp(s_{i,i+}/\tau)}{\sum_{j=1}^{2B} \mathbb{1}_{[j \neq i]} \exp(s_{i,j}/\tau)}$, $s_{i,j} = \frac{z_i \cdot z_j}{\|z_i\| \|z_j\|}$;

9 $\mathcal{L}_{NT-Xent} \leftarrow \frac{1}{2B} \sum_{i=1}^B (\ell_i + \ell_{i+})$;

 // Update parameters

10 $\phi \leftarrow \phi - \eta \nabla_\phi \mathcal{L}_{NT-Xent}$; $\theta \leftarrow \theta - \eta \nabla_\theta \mathcal{L}_{NT-Xent}$;

11 Discard g_θ ; retain pretrained encoder f_ϕ^* ;

12 **return** f_ϕ^*

and normalize it. Next, we assign $n_{c,k} = \lfloor p_{c,k} \times N_c \rfloor$ samples from class c to client k . Once all classes have been processed, each client dataset $\mathcal{D}_k$ contains samples from both classes, but in different proportions.

This replicates the data heterogeneity found in real multi-institutional medical deployments. The parameter value of $\alpha = 0.8$ yields moderate heterogeneity. The provided client distributions are similar but not identical, which constitutes a realistic and meaningful stress test for federated optimization algorithms. The global model θ_g is initialized from the Module 4 fine-tuned weights and updated over a total of $T=20$ communication rounds. Beginning at the start of each round t , the server broadcasts the global parameters $\theta_g^{(t)}$ to all K clients. Each client k initializes its local model $\theta_k \leftarrow \theta_g^{(t)}$ and retains a fixed reference copy θ_{ref} of the received global model. During $E = 5$ local training epochs, each mini-batch update minimizes the combined objective Eq. 2:

$$\mathcal{L}_k = \underbrace{\frac{1}{|\mathcal{B}_k|} \sum_{(x,y) \in \mathcal{B}_k} \text{BCE}(h_{\theta_k}(x), y)}_{\mathcal{L}_{data}} + \underbrace{\frac{\mu}{2} \|\theta_k - \theta_{ref}\|^2}_{\mathcal{L}_{prox}} \quad (2)$$

When $\mu = 0$, the proximal term disappears, so we recover FedAvg, where each client optimizes only the local data loss. If $\mu > 0$, the algorithm becomes FedProx where the square ℓ_2 -norm term $\mu/2 \|\theta_k - \theta_{ref}\|^2$ penalizes deviations from the global reference. This effectively enforces a soft constraint that reduces client drift due to heterogeneous data. When $\mu = 0.01$, the network can perform more local adaptation while still allowing some regularization. On the other hand, $\mu = 0.1$ forces greater closeness to the global model at the cost of local flexibility. After local training, all clients send their updated model parameters θ_k to the server, not the raw data. Weighted aggregation at the server is based on the client's dataset sizes, such as in Eq. 3.

$$\theta_g^{(t+1)} = \sum_{k=1}^K w_k \theta_k, \quad w_k = \frac{|\mathcal{D}_k|}{\sum_{j=1}^K |\mathcal{D}_j|} \quad (3)$$

This method allows clients with larger datasets to contribute more to the global update, given the Non-IID distribution of their dataset sizes. At the conclusion of each round, the globally updated model is assessed on

the validation set. Accuracy, AUC-ROC, and F1-score are recorded for convergence monitoring. At every round, Algorithm 4 records the logs to the blockchain ledger. The algorithm that unifies FedAvg and FedProx enables us to directly and fairly compare both strategies under the same experimental settings.

Algorithm 3: Non-IID Partitioning and Federated Training (FedAvg / FedProx)

Input: Dataset $\mathcal{D}$, clients $K = 5$, $\alpha = 0.8$, rounds $T = 20$, epochs $E = 5$, $\eta = 1 \times 10^{-4}$, $\mu \in \{0, 0.01, 0.1\}$
Output: Global model θ_g , per-round metrics

// PART A: Non-IID Data Partitioning

- 1 Initialize $\mathcal{D}_k \leftarrow \emptyset$ for all $k = 1, \dots, K$;
- 2 **for** each class $c \in \{0, 1\}$ **do**
- 3 $\mathcal{D}_c \leftarrow \{(x_i, y_i) \in \mathcal{D} : y_i = c\}$, $N_c = |\mathcal{D}_c|$;
- 4 Sample $\mathbf{p}_c \sim \text{Dirichlet}(\alpha, \dots, \alpha)$; normalize $\mathbf{p}_c$;
- 5 Shuffle $\mathcal{D}_c$; assign $n_{c,k} = \lfloor p_{c,k} \times N_c \rfloor$ samples to each client k ;
- 6 Shuffle each $\mathcal{D}_k$;

// PART B: Federated Training ($\mu = 0$: FedAvg;;
 $\mu > 0$: FedProx)

- 7 Initialize $\theta_g \leftarrow \theta_0$; // Module 4 fine-tuned weights
- 8 **for** round $t = 1$ **to** T **do**
- 9 Broadcast $\theta_g^{(t)}$ to all K clients;
- 10 **for** each client $k = 1$ **to** K **in parallel do**
- 11 $\theta_k \leftarrow \theta_g^{(t)}$; $\theta_{ref} \leftarrow \theta_g^{(t)}$; // fixed reference for proximal term
- 12 **for** local epoch $e = 1$ **to** E **do**
- 13 **for** each mini-batch $\mathcal{B}_k \subset \mathcal{D}_k$ **do**
- 14 $\mathcal{L}_{data} \leftarrow \frac{1}{|\mathcal{B}_k|} \sum_{(x,y) \in \mathcal{B}_k} \text{BCE}(h_{\theta_k}(x), y)$;
- 15 $\mathcal{L}_{prox} \leftarrow \frac{\mu}{2} \|\theta_k - \theta_{ref}\|^2$; // = 0 when $\mu = 0$ (FedAvg)
- 16 $\theta_k \leftarrow \theta_k - \eta \nabla_{\theta_k} (\mathcal{L}_{data} + \mathcal{L}_{prox})$;
- 17 Send θ_k to server;
- 18 $w_k \leftarrow \frac{|\mathcal{D}_k|}{\sum_{j=1}^K |\mathcal{D}_j|}$ for all k ;
- 19 $\theta_g^{(t+1)} \leftarrow \sum_{k=1}^K w_k \theta_k$;
- 20 Evaluate $\theta_g^{(t+1)}$; record Accuracy, AUC-ROC, F1;
- 21 Log to blockchain (Algorithm 4);
- 22 **return** $\theta_g^{(T)}$, per-round metrics

Algorithm 4 combines two complementary mechanisms that together ensure the trustworthiness and clinical interpretability of the federated cancer detection framework: blockchain-based provenance logging and Grad-CAM explainability visualization.

At the end of each communication round, after the client k completes local training and submits its updated model θ_k , a cryptographic fingerprint of the model state is computed using the SHA-256 hash function in Eq. 4:

$$h_{model} = \text{SHA-256}(\text{serialize}(\theta_k)) \quad (4)$$

A deterministic and collision-free 256-bit digest of the exact model state is produced. Even the slightest change to one model weight results in a different hash value, meaning unauthorized changes are immediately and reliably detectable. Through cryptographic chaining, the structural integrity of blockchain is maintained. In each ledger entry $\mathcal{L}_t$, with $t \geq 1$, we store the previous block hash $\mathcal{L}_{t-1}.\text{hash}$, where the first entry at $t = 1$ is assigned the null or genesis hash "0x00...00". Moreover, it refers to a chain of irreversible records in which any change to any past block invalidates all future blocks. Every entry $\mathcal{L}_t$ contains the communication round t the client identifier k a UTC timestamp the hash of the model h_{model} the hash of the previous block h_{prev} the accuracy, area under the curve and the standard F1 score of the client update all indicated with acc , auc , $f1$ respectively and finally the size of

the local training set with $|\mathcal{D}_k|$. To seal the entry, the block computes its own block hash. It uses the function $\mathcal{L}_t.\text{hash} = \text{SHA-256}(\text{serialize}(\mathcal{L}_t))$ to compute the block hash. Further, the integrity of the chain is verified before it is appended to the persistent JSON ledger. Over $T = 20$ rounds with $K = 5$ clients, it generates a complete audit trail of $T \times K = 100$ immutable ledger entries per training run for compliance and forensic verification.

Gradient-weighted Class Activation Mapping (Grad-CAM) is applied to the final global federated model to generate spatial heatmaps that identify the regions of a histopathological image that most strongly influenced the model’s classification decision. A standard forward pass yields the prediction $\hat{y}$ and captures the feature maps $\{A^k\}_{k=1}^K$ from the final convolutional layer `encoder.features[-1]`. The class score y^c for target class c is back-propagated to obtain gradients $\frac{\partial y^c}{\partial A^k}$, which quantify the sensitivity of the class prediction to each spatial location in each feature map. Global average pooling is applied to compute a scalar importance weight per feature map as shown in Eq. 5:

$$\alpha_k^c = \frac{1}{Z} \sum_{i=1}^u \sum_{j=1}^v \frac{\partial y^c}{\partial A_{i,j}^k}, \quad Z = u \times v \quad (5)$$

The Grad-CAM heatmap is then computed as a weighted combination of the feature maps, followed by ReLU activation to retain only the regions that positively contribute to the target class prediction using Eq. 6:

$$\mathcal{M} = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c \cdot A^k \right) \quad (6)$$

The heatmap $\mathcal{M}$ is upsampled via bilinear interpolation to the input resolution of 224×224 pixels, min-max normalized to $[0, 1]$, and converted to a false-color representation using the JET colormap, where blue indicates low activation and red indicates high activation.

3.1 Data Preprocessing

This paper used two histopathological image-based datasets: PatchCamelyon (PCam) and BreakHis. We applied a systematic preprocessing approach to both datasets to ensure quality, consistency, and alignment with the proposed federated learning framework.

3.1.1 PatchCamelyon Dataset

The PCam dataset <https://huggingface.co/datasets/zacharielegault/PatchCamelyon> contains 327,680 histopathology patches, each measuring 96×96 pixels, and labeled as either normal tissue (class 0) or tumor tissue (class 1). The dataset was sourced from the HuggingFace repository, which has predefined splits for training (262,144 samples), validation (32,768 samples), and testing (32,768 samples). We resize each image from 96×96 to 224×224 pixels using bilinear interpolation to meet the input size requirements of the EfficientNet-B0 model. The models were trained on images undergoing a stochastic data augmentation pipeline to improve generalization and prevent overfitting. The images in the datasets are transformed using random horizontal and vertical flips, and color jitter with a disturbance factor of 0.2 for brightness, contrast, and saturation. Each image was converted to a tensor, and its values were normalized across the three channels: red (R), green (G), and blue (B). For each channel, the mean used was $[0.5, 0.5, 0.5]$ and the standard deviation was $[0.5, 0.5, 0.5]$. For the validation and test splits, only resizing and normalization were performed, with no augmentations.

3.1.2 BreakHis Dataset

The BreakHis dataset <https://www.kaggle.com/datasets/ambarish/breakhis/data> comprises microscopy images of breast tumor tissues, categorized as benign or malignant. This dataset was organized into four federated client directories, with each directory containing separate training and testing subdirectories for each class. The configuration file made available to us was used to allocate images to clients, and Fold 1 was used for testing across all experimental configurations. The BreakHis images were resized to 224×224 pixels. Data augmentation was performed as a preprocessing step on the network’s training images using random horizontal flipping and random rotations up to 10° . In our centralized training experiments, we normalize ImageNet with mean = $[0.485, 0.456, 0.406]$, std = $[0.229, 0.224, 0.225]$. In federated training, however, we use a mean of $[0.5, 0.5, 0.5]$ and a std of $[0.5, 0.5, 0.5]$ for consistency. The entire training pool was split into training and validation subsets using a 90:10 random split to monitor performance across federated communication rounds.

Algorithm 4: Blockchain Provenance and Grad-CAM Explainability

Input: [Blockchain]: Model θ_k , client k , round t , metrics $\{acc, auc, f1\}$ [Grad-CAM]: image x , class c , target layer L
Output: Immutable ledger entry $\mathcal{L}_t$; Grad-CAM overlay $\mathcal{M}_{final}$
// PART A: Blockchain Provenance
1 $h_{model} \leftarrow \text{SHA-256}(\text{serialize}(\theta_k))$; // cryptographic model fingerprint
2 **if** $t = 1$ **then**
3 | $h_{prev} \leftarrow "0x00...00"$; // genesis block
4 **else**
5 | $h_{prev} \leftarrow \mathcal{L}_{t-1}.\text{hash}$; // chain linkage
6 $\mathcal{L}_t \leftarrow \{\text{round}:t,;$
7 $\text{client}:k,;$
8 $\text{timestamp}:\tau_{UTC},;$
9 $\text{model_hash}:h_{model},;$
10 $\text{prev_hash}:h_{prev},;$
11 $acc,;$
12 $auc,;$
13 $f1,;$
14 $|\mathcal{D}_k| \}$;
15 $\mathcal{L}_t.\text{hash} \leftarrow \text{SHA-256}(\text{serialize}(\mathcal{L}_t))$; // seal the block
16 **if** $t > 1$ **then**
17 | $\text{Verify } \mathcal{L}_t.\text{prev_hash} = \mathcal{L}_{t-1}.\text{hash}$; // integrity check
18 Append $\mathcal{L}_t$ to model_ledger.json;
// PART B: Grad-CAM Explainability
19 Forward pass: $\hat{y} \leftarrow h_\theta(x)$; extract $\{A^k\}_{k=1}^K$ from layer L ;
20 Backpropagate y^c ; compute $\frac{\partial y^c}{\partial A^k}$ for all k ;
21 **for each** feature map k **do**
22 | $\alpha_k^c \leftarrow \frac{1}{Z} \sum_{i,j} \frac{\partial y^c}{\partial A_{i,j}^k}$; // global average pooling, $Z = u \times v$
23 $\mathcal{M} \leftarrow \text{ReLU}(\sum_{k=1}^K \alpha_k^c \cdot A^k)$; // positive contributions only
24 $\mathcal{M} \leftarrow \text{Resize}(\mathcal{M}, 224 \times 224)$; $\mathcal{M} \leftarrow \frac{\mathcal{M} - \min \mathcal{M}}{\max \mathcal{M} - \min \mathcal{M}}$; // upsample + normalize
25 $\mathcal{M}_{final} \leftarrow 0.6 \times x + 0.4 \times \text{JetColormap}(\mathcal{M})$; // blend with original image
26 **return** Ledger entry $\mathcal{L}_t$; Grad-CAM overlay $\mathcal{M}_{final}$

Table 1 provides the properties of both datasets. The PCam dataset provides a balanced set of 327,680 histopathological patches, pre-divided into training, validation, and test splits, making it well-suited for evaluating federated learning at scale. The BreaKHis dataset presents a more clinically realistic and challenging scenario with two diagnostic categories, benign and malignant, organized across multiple magnification levels. All images were resized to 224×224 pixels to conform to the EfficientNet-B0 input requirements, and standard data augmentation and normalization pipelines were applied. PCam provides over 262,000 training samples with a predefined validation split of 32,768, facilitating stable federated training across five clients. BreaKHis, with 20,466 training images, presents a more challenging scenario where data scarcity per client is a realistic concern.

Table 1. Dataset Statistics for PCam and BreaKHis

Dataset	Split	Samples	Classes	Image Size
PatchCamelyon (PCam)	Train	262,144	2 (Normal/Tumor)	$96 \times 96 \rightarrow 224 \times 224$
	Validation	32,768		
	Test	32,768		
BreaKHis	Train	20,466	2 (Benign/Malignant)	224×224
	Test	7,909		

3.1.3 Federated Data Partitioning

To simulate realistic federated learning environments, the training data from both datasets were distributed across clients using a Dirichlet-based Non-IID partitioning strategy with a concentration parameter $\alpha = 0.8$. This approach introduces controlled label heterogeneity among clients, reflecting the data imbalance often found in healthcare settings, where different hospitals serve distinct patient populations. For the PCam dataset, training data was partitioned among $K = 5$ clients, with a minimum threshold of 5,000 samples per client. The BreakHis dataset was similarly distributed across $K = 4$ clients, with each client having at least 3,000 samples, ensuring adequate data availability during local training.

3.1.4 Self-Supervised Preprocessing (SimCLR)

In the self-supervised contrastive pretraining Module (Module 3), a specialized augmentation pipeline was developed for both datasets. All images were randomly resized, horizontally flipped, color jittered (with a probability of 0.8), and randomly converted to grayscale (with a probability of 0.2). During SimCLR training, images are passed through the augmentation pipeline twice to obtain two augmented views, which form a positive pair. To save computational time, the images in this Module were resized to 96×96 pixels and processed in batches of 128. This will remove unwanted information while still containing useful information.

3.2 Baseline Supervised Training (Model Architecture and Training Configuration and Evaluation Protocol)

Before beginning federated learning, we performed a centralized supervised training experiment on both datasets to establish performance baselines. The performance drop resulting from federated optimization in Non-IID data distributions is compared with this baseline. We used EfficientNet-B0 as a backbone architecture for baseline training on the PCam and BreakHis datasets. We chose EfficientNet-B0 because it jointly optimizes the network depth, width, and input resolution to deliver good accuracy with fewer parameters than other models. We initialized the model with weights pre-trained on ImageNet to benefit from transfer learning, aiding feature extraction from histopathological images, even with limited specific training data. For binary classification, we replaced the final fully connected layer of the EfficientNet-B0 classifier with a single-output linear layer followed by a sigmoid activation. This setup maps the extracted features to a tumor probability score $p \in [0, 1]$. We optimized the model using the Adam optimizer with a learning rate of $\eta = 1 \times 10^{-4}$ and a weight decay of 1×10^{-4} to regularize the training and help prevent overfitting. The training objective was defined using the Binary Cross-Entropy with Logits loss (BCEWithLogitsLoss) Eq. 7:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)] \quad (7)$$

where $y_i \in \{0, 1\}$ indicates the true label, and $\hat{y}_i$ represents the predicted probability for sample i . We trained the model for 10 epochs with a batch size of 32. For the BreakHis dataset, we applied early stopping with a patience of 2 epochs based on validation AUC to prevent overfitting, particularly given the dataset’s smaller size. All experiments were conducted in a GPU-enabled environment using mixed-precision training to improve computational speed.

We evaluated model performance on the held-out test set after each training run using three metrics: classification accuracy, area under the receiver operating characteristic curve (AUC-ROC), and sensitivity (true positive rate). The BreakHis baseline training used the same configuration across all four client directories, monitoring performance via training and validation loss curves, ROC curves, and confusion matrices to ensure reliable, consistent results.

3.3 Self-Supervised Pretraining (SimCLR)

Self-supervised pretraining was performed before supervised fine-tuning to learn meaningful visual representations from unlabeled histopathological images. This Module is particularly valuable in medical imaging, where labeled data is scarce due to the expertise required for annotation, while large quantities of unlabeled images are readily available. The Simple Framework for Contrastive Learning of Visual Representations (SimCLR) was adopted for this purpose, as it has demonstrated strong representation-learning capabilities without requiring labeled supervision. The SimCLR framework consists of two components: an encoder network f_ϕ and a projection head g_θ . EfficientNet-B0, pretrained on ImageNet, was used as the encoder backbone to extract high-dimensional feature representations from input images. The original classifier head of EfficientNet-B0 was replaced with an identity mapping, allowing the encoder to output raw feature vectors. A two-layer multilayer perceptron (MLP) projection head g_θ was appended to map the encoder outputs to a 128-dimensional embedding space, where the contrastive loss is computed. The projection head serves only as an auxiliary training component and is discarded after pretraining.

3.3.1 Data Augmentation Strategy

SimCLR relies on generating two distinct augmented views of the same image to form positive pairs for contrastive learning. For each input image x_i , the augmentation pipeline $\mathcal{T}$ was applied twice independently to produce two views $\tilde{x}_i^{(1)}$ and $\tilde{x}_i^{(2)}$. The augmentation pipeline comprised random resized cropping with a scale range of (0.2,1.0), random horizontal flipping, color jitter (brightness, contrast, saturation, and hue perturbations applied with probability 0.8), and random grayscale conversion applied with probability 0.2. These stochastic transformations ensure that the two views exhibit sufficient visual variation while still originating from the same underlying image, compelling the model to learn augmentation-invariant representations. To reduce computational cost, images were resized to 96×96 pixels during this Module, with a batch size of 128 to provide sufficient negative pairs for effective contrastive learning.

3.3.2 Contrastive Learning Objective

The normalized temperature-scaled cross-entropy (NT-Xent) loss was employed as the training objective. For each image x_i in a batch of size B , the encoder and projection head produce normalized embeddings $z_i^{(1)} = g_\theta(f_\phi(\tilde{x}_i^{(1)}))$ and $z_i^{(2)} = g_\theta(f_\phi(\tilde{x}_i^{(2)}))$. The cosine similarity between embeddings is computed as 8:

$$s_{i,j} = \frac{z_i^{(1)} \cdot z_j}{\|z_i^{(1)}\| \cdot \|z_j\|} \quad (8)$$

The contrastive loss for each sample i is then defined as Eq. 9:

$$\ell_i = -\log \frac{\exp(s_{i,i^+}/\tau)}{\sum_{j=1}^{2B} \mathbb{1}_{[j \neq i]} \exp(s_{i,j}/\tau)} \quad (9)$$

where i^+ denotes the index of the positive pair (i.e., the second augmented view of the same image), $\tau = 0.5$ is the temperature parameter controlling the sharpness of the similarity distribution, and $\mathbb{1}_{[j \neq i]}$ is an indicator function that excludes the anchor sample itself from the denominator. The batch-level loss is computed symmetrically over all samples as Eq. 10:

$$\mathcal{L}_{NT-Xent} = \frac{1}{2B} \sum_{i=1}^B (\ell_i + \ell_{i^+}) \quad (10)$$

This formulation encourages the model to bring positive pairs closer together in the embedding space while pushing negative pairs apart, corresponding to augmented views of the same image within the same batch.

3.3.3 Training Configuration

The SimCLR pretraining was conducted for 50 epochs using the Adam optimizer with a learning rate of $\eta = 1 \times 10^{-3}$. Pretraining was performed independently on both the PCam and BreakHis datasets using their respective unlabeled training sets. Upon completion of pretraining, the projection head g_θ was discarded, and the pretrained encoder f_ϕ^* was retained as a fixed feature extractor for the subsequent supervised fine-tuning Module. The encoder weights learned through contrastive pretraining capture rich, transferable visual representations that enable effective classification even when labeled data is limited.

3.4 Centralized Supervised Fine-Tuning

Following self-supervised pretraining via SimCLR, a centralized supervised fine-tuning Module was conducted to adapt the learned representations for binary cancer classification. This Module bridges the gap between unsupervised representation learning and the downstream diagnostic task by introducing labeled supervision on top of the frozen SimCLR encoder. The fine-tuned model subsequently serves as the initialization point for all federated learning experiments. A FineTunedModel was constructed by loading the pretrained EfficientNet-B0 encoder weights from the SimCLR pretraining Module for the PCam and BreakHis datasets, respectively. During pretraining, the original classifier head of EfficientNet-B0 was replaced with an identity mapping, and a new linear classification head was attached to the encoder. Specifically, a fully connected layer of dimensions $\mathbb{R}^{1280} \rightarrow \mathbb{R}^2$ was used for multi-class output with cross-entropy loss on the PCam dataset, while a single-output linear layer $\mathbb{R}^{1280} \rightarrow \mathbb{R}^1$ with binary cross-entropy loss was used for the BreakHis dataset. The architecture can be formally expressed as Eq. 11:

$$\hat{y} = W \cdot f_{\phi}^*(x) + b \quad (11)$$

where f_{ϕ}^* denotes the frozen SimCLR pretrained encoder, W and b are the trainable parameters of the classification head, and $\hat{y}$ is the predicted class logit.

3.4.1 Encoder Freezing Strategy

A key design decision in this Module was to freeze the encoder parameters completely during fine-tuning, i.e., all parameters of f_{ϕ}^* were set with `requires_grad = False`. Only the classification head parameters were updated during backpropagation. This strategy was adopted for two reasons. First, it preserves the rich, generalizable representations learned during contrastive pretraining from being overwritten by a small labeled dataset. Second, and critically for the federated learning Modules that follow, a frozen encoder ensures that only the lightweight classifier head needs to be communicated between clients and the server during federated aggregation, substantially reducing communication overhead.

3.4.2 Training Configuration

The fine-tuning was performed using the AdamW optimizer with a learning rate of $\eta = 1 \times 10^{-3}$ and a weight decay of 1×10^{-4} , applied exclusively to the classifier head parameters. Training was conducted for up to 10 epochs with a batch size of 32. Early stopping with a patience of 2 epochs was applied based on the validation AUC-ROC score to prevent overfitting, retaining the model state that achieved the best validation performance. Mixed-precision training with GradScaler was used to accelerate GPU computation while maintaining numerical stability. The training objective for the PCam dataset was the cross-entropy loss Eq. 12:

$$\mathcal{L}_{CE} = -\frac{1}{N} \sum_{i=1}^N \sum_{c=0}^{C-1} y_{i,c} \log(\hat{p}_{i,c}) \quad (12)$$

while for the BreaKHis dataset, the binary cross-entropy with logits loss was used Eq. 13:

$$\mathcal{L}_{BCE} = -\frac{1}{N} \sum_{i=1}^N [y_i \log(\hat{p}_i) + (1 - y_i) \log(1 - \hat{p}_i)] \quad (13)$$

where y_i is the ground truth label, $\hat{p}_i$ is the predicted probability, N is the total number of samples, and C denotes the number of classes.

3.4.3 Evaluation

Model performance was evaluated on the held-out test set using accuracy, AUC-ROC, and F1-score. To further validate the quality of the learned representations, we applied t-SNE to the encoder feature vectors extracted from test set images. A well-separated t-SNE visualization with distinct clusters for normal and tumor classes confirms that the SimCLR encoder has learned discriminative features that generalize effectively to labeled data. The fine-tuned model weights, specifically the encoder f_{ϕ}^* and the trained classifier head, were saved and used to initialize the global model for all subsequent federated learning experiments, ensuring a consistent and informed starting point across all communication rounds.

3.5 Non-IID Client Data Splitting

In practice, the data exchanged between learning institutions in federated learning programs is often not IID. Different patient populations, imaging devices with varying characteristics, and diagnostic protocols across hospitals and clinical centres lead to divergent local data distributions. To impose a more realistic scenario onto this setting, we apply a Non-IID data partitioning strategy. The training data is allocated among federated clients based on the Dirichlet distribution for both datasets.

3.5.1 Partitioning Strategy

The Dirichlet distribution with concentration parameter $\alpha = 0.8$ was used to generate heterogeneous label distributions across clients. For each class $c \in \{0, 1\}$, a proportion vector $\mathbf{p}_c = (p_{c,1}, p_{c,2}, \dots, p_{c,K})$ was sampled as Eq. 14:

$$\mathbf{p}_c \sim \text{Dirichlet}(\alpha, \alpha, \dots, \alpha) \quad (14)$$

where each component $p_{c,k}$ represents the fraction of class c samples allocated to client k , and K denotes the total number of clients. The sampled proportions were normalized to ensure $\sum_{k=1}^K p_{c,k} = 1$, and the actual number of samples assigned to each client for class c was computed as $n_{c,k} = \lfloor p_{c,k} \times N_c \rfloor$, where N_c denotes the total number of samples belonging to class c in the training pool. Each class was processed independently, and the resulting per-class sample sets were combined to form each client’s local dataset $\mathcal{D}_k$. The final client datasets were randomly shuffled to mix class samples before training.

The concentration parameter $\alpha = 0.8$ was selected to simulate moderate Non-IID heterogeneity. Lower values of α create increasingly severe label imbalance across clients, approaching a pathological Non-IID case, whereas larger values converge to a uniform IID distribution. The value of $\alpha = 0.8$ therefore represents a realistic federated setting with client data features that are similar but not identical, as commonly found in multi-institutional medical imaging data.

3.5.2 PCam Dataset Partitioning

For the PatchCamelyon dataset, the training pool distributed across $K = 5$ clients comprised 262,144 samples. An iterative resampling mechanism enforced a safe partitioning procedure. If any client obtained fewer than 5000 samples after a Dirichlet draw, then the sampling was repeated with a new random seed until all clients satisfied the condition. This condition ensures that no client is deprived of data during local training, thereby preventing local model degeneration and unstable gradient updates. The client datasets that we generated have varying sizes and classes. Some clients have more tumor samples, while others have more normal tissue, reflecting realistic inter-institutional variability.

3.5.3 BreaKHis Dataset Partitioning

To create the BreaKHis dataset, the pooled training set from all four client directories was first combined into a shared pool, and labels were selected to enable a class-wise Dirichlet split. The same Dirichlet-based procedure was used for redistribution as well, with $K = 4$ and $\alpha = 0.8$. A lower limit of 3,000 samples per client was applied, in line with the smaller overall size of the BreaKHis dataset compared to PCam. After partitioning the dataset, the data for each client was split using a PyTorch Subset. Moreover, a DataLoader with a batch size of 32 was created to load the datasets for subsequent rounds of federated training. The Non-IID distribution partitioning scheme is a vital stress test of the federated optimization algorithms tested in this work. FedAvg is theoretically guaranteed to converge to the optimal global model under IID conditions. Nonetheless, non-IID distributions cause local training objectives to conflict across clients, leading to local models drifting in different directions and potentially destabilizing global convergence. By introducing controlled heterogeneity via Dirichlet sampling, this Module facilitates a careful examination of both FedAvg and FedProx in realistic settings, thereby motivating the use of proximal regularization in FedProx to counter client drift.

3.6 Federated Learning

Federated learning was employed as the core privacy-preserving training paradigm in the proposed framework. Rather than aggregating raw patient data at a central server, each client trains a local model on its private dataset and shares only the model parameters with the server for aggregation. This section provides the two federated optimization strategies evaluated in this study: FedAvg, the baseline, and FedProx, the heterogeneity-aware variant, followed by a comprehensive evaluation protocol applied across both datasets.

3.6.1 Baseline Federated Learning (FedAvg)

The baseline federated optimization algorithm implemented the Federated Averaging (FedAvg). The model is trained for $T = 20$ communication rounds. At the start of every round t of the algorithm, the server sends the current global model parameters $\theta_g^{(t)}$ to all K participating clients. Every client k initializes its local model with the received global parameters and performs $E = 5$ epochs of supervised training on its private local dataset $\mathcal{D}_k$ using cross-entropy loss and the AdamW optimizer with learning rate $\eta = 1 \times 10^{-3}$.

Crucially, the encoder f_ϕ^* pretrained with SimCLR was kept frozen throughout all local training rounds, with gradients disabled for all encoder parameters. During local training, only the lightweight linear classification head was updated, resulting in significantly fewer parameters communicated per round and faster convergence. To maximize GPU utilization, automatic mixed precision (AMP) via GradScaler is applied to all local training steps.

Once all clients finish their on-edge training, the updated model parameters $\{\theta_k\}_{k=1}^K$ are sent to the server. The global model is updated using the weighted aggregation based on the dataset sizes of the clients on the server as Eq. 15:

$$\theta_g^{(t+1)} = \sum_{k=1}^K w_k \theta_k, \quad \text{where} \quad w_k = \frac{|\mathcal{D}_k|}{\sum_{j=1}^K |\mathcal{D}_j|} \quad (15)$$

where $|\mathcal{D}_k|$ denotes the number of training samples available at client k . Through this weighted averaging, clients with larger local data (non-IID partitioning) contribute more to the global model. Thus, the global model can reflect different client dataset sizes (non-IID). The global model $\theta_g^{(t+1)}$ is then updated and sent to all clients at the beginning of the next communication round. This is done for $T = 20$ rounds.

3.6.2 Heterogeneity-Aware Federated Learning (FedProx)

While FedAvg performs well under moderate data heterogeneity, it does not explicitly constrain the degree to which local models diverge from the global model during local training. Under Non-IID conditions, clients with skewed local label distributions may update their models in conflicting directions, a phenomenon known as client drift, which can slow convergence or degrade global model quality. To address this, Federated Proximal (FedProx) was implemented as a heterogeneity-aware alternative to FedAvg. FedProx introduces a proximal regularization term into each client's local optimization objective, penalizing excessive deviation from the current global model. The modified local loss function for client k is defined as Eq. 16:

$$\mathcal{L}_k^{\text{FedProx}} = \mathcal{L}_{\text{data}} + \frac{\mu}{2} \|\theta_k - \theta_{\text{global}}\|^2 \quad (16)$$

where $\mathcal{L}_{\text{data}}$ is the standard cross-entropy loss computed on the local data, θ_{global} is the fixed copy of the global model received at the beginning of the communication round, θ_k are the current local model parameters, and μ is the proximal regularization coefficient that controls the strength of the penalty. The squared ℓ_2 -norm term $\|\theta_k - \theta_{\text{global}}\|^2$ acts as a soft constraint that pulls the local model back toward the global model during optimization, preventing local updates from drifting too far in a direction that is only beneficial for that client's local distribution.

In practice, the proximal term was computed only over the trainable parameters of the classifier head, since the encoder was frozen. Two values of the proximal parameter were evaluated, $\mu \in \{0.01, 0.1\}$, to study the effect of regularization strength on convergence and final performance. A smaller value of $\mu = 0.01$ allows greater local adaptation to client-specific data, while $\mu = 0.1$ enforces closer alignment with the global model, prioritizing global consistency over local fit. The server-side aggregation in FedProx is identical to FedAvg, using the same weighted averaging formula in Equation (3), ensuring a fair comparison between the two algorithms. All other hyperparameters, including the number of communication rounds ($T = 20$), local epochs ($E = 5$), learning rate ($\eta = 1 \times 10^{-3}$), and batch size (32), were kept identical across both FedAvg and FedProx experiments.

The global model was evaluated on a held-out validation set after each communication round to monitor training progress and detect convergence. After completing all $T = 20$ rounds, the final global model was evaluated on the test set using three performance metrics: classification accuracy, area under the receiver operating characteristic (ROC) curve (AUC-ROC), and F1-score. These metrics were computed as shown in Eq. 17, 18:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN} \quad (17)$$

$$\text{F1-score} = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}} \quad (18)$$

where TP , TN , FP , and FN denote true positives, true negatives, false positives, and false negatives, respectively. AUC-ROC was computed using the predicted probability scores output by the sigmoid or softmax activation applied to the final classifier logits.

Per-client evaluation was also performed by independently running the final global model on each client's local test data, enabling analysis of performance variance across clients with different local data distributions. This per-client evaluation indicates whether the global model generalizes uniformly across heterogeneous clients or exhibits performance divergence attributable to label imbalance. To visually compare the convergence behavior of FedAvg and FedProx, the validation accuracy and loss curves across communication rounds were plotted. To visualize the impacts of the proximal regularization strength on the training dynamics on PCam and BreakHis, separate convergence curves were obtained for FedProx with $\mu = 0.01$ and $\mu = 0.1$.

3.7 Blockchain Provenance

A blockchain-influenced logging system was implemented to provide transparency, accountability, and auditability for the federated learning process within the framework. This system offers a secure, tamper-proof audit trail of all model updates made by each client during every communication round. A provenance system is essential in sensitive healthcare environments to comply with the law, conduct forensic investigations, and generate trust in institutional AI-assisted diagnoses.

3.7.1 Model Hashing

Upon completing a local training round, the state dictionary of client k with model parameters θ_k will be serialized and hashed using the SHA-256 hash function (Secure Hash Algorithm, 256-bit). In particular, all parameter tensors in the state dictionary are first converted to a numerical list and then serialized to a UTF-8 string. The application of the SHA-256 hash function yields a 256-bit fingerprint of the model, Eq. 19.

$$h_{model} = \text{SHA-256} \left(\text{encode} \left(\left\{ k : \theta_k^{(l)}.tolist() \ \forall l \in \mathcal{L} \right\} \right) \right) \quad (19)$$

In this equation, the set $\mathcal{L}$ consists of all the named parameter layers of the model, $\theta_k^{(l)}$ refers to the parameter tensor of layer l for client k , and $\text{encode}(\cdot)$ denotes UTF-8 string encoding. The hash h_{model} , which is produced, is deterministic and has a fixed length, which is a hexadecimal digest. In other words, it is a unique fingerprint of the model at that point in training. Even a minuscule change to any model weight produces a completely different hash value, making it highly sensitive to unauthorized or accidental changes to the model.

3.7.2 Ledger Entry Construction

Upon computing the hash, a structured ledger record is created for each client update. The entry encompasses comprehensive metadata capturing the federated learning iteration, which includes the communication round number, the identifier of the client, UTC timestamp for the update, model hash h_{model} , the proximal regularization term μ for local training, local dataset size $|\mathcal{D}_k|$, and the validation performance (accuracy, AUC-ROC and F1-score) perceiving on the global model at that round. This rich metadata structure can be formally expressed by Eq. 20.

$$\mathcal{L}_t^{(k)} = \left\{ \begin{array}{l} round : t, client_id : k, timestamp : \tau, \\ model_hash : h_{model}, mu : \mu, val_acc : acc_t \end{array} \right\} \quad (20)$$

In this context, t is the communication round, k is the client ID, τ is the UTC timestamp, and acc_t is the validation accuracy after round t . Each entry in the model ledger is saved in a persistent JSON file (model_ledger.json for the PCam dataset and Module7_model_ledger.json for the BreakHis dataset), which continues to grow throughout training to maintain a complete, chronological record.

3.7.3 Tamper-Evidence and Chain Integrity

The records in the ledger are stored in chronological order and summed across all communication rounds and clients. The model hash for each entry corresponds to a client update made in that round. Therefore, any alteration of a previously submitted model update will produce a different hash, which will be promptly detected in a ledger audit. In federated health care environments where resources may be limited, this approach is lightweight. It does not require a fully distributed peer-to-peer blockchain infrastructure to provide evidence of tampering. The blockchain provenance mechanism was applied uniformly across all federated optimization strategies in the paper, including FedAvg and FedProx, with $\mu \in \{0.01, 0.1\}$. It was also applied to PCam and BreakHis datasets. The BreakHis experiments used separate ledger files for each value of μ to allow independent audits on each configuration.

3.7.4 Ledger-Based Performance Analysis

In addition to storing training data for security purposes, the ledger was a structured data store. This framework would help determine whether the training was effective. In all federated training rounds, the ledger was checked to obtain per-round validation accuracy, aggregated across clients. The paper then obtained the mean accuracy and proxy validation loss (calculated as $1 - \text{accuracy}$) for each communication round. All these metrics are visualized as convergence curves over communication rounds T for FedAvg and FedProx, and they are clearly transparent and verifiable on the ledger. The ledger's dual function as a cryptographic audit trail and structured performance log illustrates how blockchain-inspired provenance mechanisms can be effectively embedded in federated learning pipelines to achieve security and scientific reproducibility goals simultaneously.

3.8 Explainability

The addition of Gradient-weighted Class Activation Mapping (Grad-CAM) improves clinical trust and interpretability of the federated cancer detection framework. Even though deep learning models often provide high predictive power, they often face black-box criticism because it is impossible to explain their decision-making to clinicians. Grad-CAM mitigates this issue by generating spatial heatmaps that highlight the most influential regions of an image in determining the model's classification decision, allowing pathologists to confirm that the model focuses on clinically relevant tissue structures rather than extraneous artifacts.

Grad-CAM was applied to the final global federated models trained with FedProx, using proximal parameters $\mu \in \{0.01, 0.1\}$, for both the PCam and BreakHis datasets. For each μ value, the saved global model weights (Module7_global_model_mu_{\mu}.pth) were loaded into the GlobalModel structure and set to evaluation mode. The target convolutional layer for gradient extraction was set to the final feature block of the EfficientNet-B0 encoder, accessed via model.encoder.features[-1]. This layer was selected due to its ability to capture the highest-level semantic features learned by the network, which are most directly associated with the final classification decision.

The Grad-CAM algorithm was executed using a hook-based mechanism. Two hooks were registered on the target layer: a forward hook that captures the layer's output feature maps $A = \{A^k\}_{k=1}^K$ during the forward pass, and a full backward hook that captures the gradients $\frac{\partial y^c}{\partial A^k}$ that flow back through the layer during backpropagation. For a specific input image x and target class c , the forward pass was performed with gradient computation enabled via the enable_grad=True flag, and the class score y^c was backpropagated to obtain the gradients with respect to each feature map. The importance weight for each feature map k is computed by applying global average pooling to the corresponding gradients Eq. 21:

$$\alpha_k^c = \frac{1}{Z} \sum_{i=1}^u \sum_{j=1}^v \frac{\partial y^c}{\partial A_{i,j}^k} \quad (21)$$

where $Z = u \times v$ is the total number of spatial locations in the feature map, and u and v denote its height and width, respectively. The Grad-CAM heatmap is then computed as a weighted linear combination of the feature maps:

$$\mathcal{M}_{Grad-CAM} = \text{ReLU} \left(\sum_{k=1}^K \alpha_k^c \cdot A^k \right) \quad (22)$$

The ReLU activation retains only the feature map regions that positively contribute to the target class prediction, discarding regions associated with negative evidence. The resulting heatmap is then normalized to the range $[0, 1]$ using min-max normalization:

$$\mathcal{M}_{Grad-CAM} = \frac{\mathcal{M}_{Grad-CAM} - \min(\mathcal{M}_{Grad-CAM})}{\max(\mathcal{M}_{Grad-CAM}) - \min(\mathcal{M}_{Grad-CAM}) + \epsilon} \quad (23)$$

where $\epsilon = 1 \times 10^{-8}$ is a small constant added for numerical stability to prevent division by zero.

The normalized heatmap was upsampled to match the original input image resolution of 224×224 pixels using bilinear interpolation via OpenCV's resize function. The resulting heatmap was created using the JET colormap to provide insight into activation values. Specifically, blue indicates lower activation, while red indicates higher activation. In essence, this allows for an assessment of regions of greater discrimination. The last rendering was generated by linearly interpolating between the heatmap colormap and the original image.

$$\mathcal{M}_{final} = \frac{0.6 \times x_{norm} + 0.4 \times \mathcal{M}_{color}}{\max(0.6 \times x_{norm} + 0.4 \times \mathcal{M}_{color})} \quad (24)$$

where x_{norm} is the min-max normalized original image and $\mathcal{M}_{color}$ is the JET colormap-transformed heatmap. The blending weights of 0.6 and 0.4 were chosen to keep the original image clearly visible while ensuring that the heatmap activations are prominently overlaid for inspection.

To deliver a balanced and representative explainability analysis, Grad-CAM visualizations were generated for one representative test image per class in each dataset, for each value of μ . For the PCam dataset, selected one normal tissue (class 0) and one tumor tissue (class 1), and for the BreakHis dataset, selected one benign sample and

a malignant sample. By examining the model’s attention patterns within each class, we can see how it differentiates between normal and cancerous tissues, assuming these tissues are diagnostic of their respective classes. For each FedProx model trained with $\mu = 0.01$ and $\mu = 0.1$, the process was performed independently to visually compare the effect of the proximal regularization strength on the spatial focus of the learned representation.

The Grad-CAM visualizations perform several important functions in the framework. One, they provide clinicians with interpretability by visualising which spatial regions of the histopathological image influence the model’s prediction. This ensures that the model predictions are based on diagnostically relevant tissue structures instead of staining artefacts or background regions. Secondly, they help diagnose errors by highlighting instances where the model focuses on clinically incorrect areas. This observation raises a concern about the quality of the training data or potential model bias, warranting further examination. Third, they enable regulatory compliance, as regulatory bodies call for documentation of explainability for clinical deployment approval of medical AI. The framework can produce a cancer detection system with transparent, clinically interpretable output that delivers high predictive performance and is practically useful by applying Grad-CAM, a systematic post-training analysis step applied to all federated model configurations.

4 Results

This part of the paper provides a detailed empirical evaluation of the proposed framework on complete cancer. The experiments were carried out in an NVIDIA GPU-enabled environment using PyTorch with mixed-precision training. The system underwent an assessment across eight sequential Modules using two publicly available histopathology image datasets: PatchCamelyon (PCam) and BreakHis. As each Module is processed, results are quantified and analyzed; thereby demonstrating the progressive evolution of the framework from centralized baseline training to privacy-preserving federated deployment with blockchain provenance and Grad-CAM explainability.

4.1 Evaluation Measures and Environment

To enable reproducibility and transparency, we specify in this subsection the complete hardware and software environment used to execute all experiments, followed by a comprehensive summary of all model architecture parameters and training hyperparameters used in the framework.

Table 2 summarizes the hardware and software configuration used throughout all experimental Modules. All experiments were conducted in a cloud-based environment equipped with an NVIDIA Tesla T4 GPU (16 GB VRAM), an Intel Xeon CPU (2 cores), and 12 GB of RAM. The implementation was developed using Python 3.12.12 with PyTorch 2.10.0 compiled with CUDA 12.8.

Table 2. Experimental Environment Configuration

Component	Specification
GPU	NVIDIA Tesla T4 (16 GB VRAM)
CPU	Intel Xeon (2 cores)
RAM	12 GB
Python	3.12.12
PyTorch	2.10.0 (CUDA 12.8)
torchvision	0.25.0 (CUDA 12.8)

Table 3 provides a comprehensive summary of all model architecture specifications and training hyperparameters used across every modules of the proposed framework. The backbone architecture for all Modules is EfficientNet-B0, selected for its favourable trade-off between parameter efficiency and classification performance. EfficientNet-B0 has approximately 5.3 million parameters in its convolutional feature-extraction layers, with a final feature dimension of 1,280 before the classification head. In Module 2 (baseline supervised training), the full model is fine-tuned with ImageNet pretrained weights using a single linear output neuron for binary classification with BCE loss and AdamW optimizer at a learning rate of 1×10^{-4} and weight decay of 1×10^{-4} for 10 epochs with a batch size of 32. In Module 3 (SimCLR pretraining), the classification head is replaced with a two-layer MLP projection head that maps the 1,280-dimensional encoder output to a 128-dimensional contrastive embedding space via an intermediate dimension of 256, adding approximately 360,000 parameters. Pretraining runs for 10 epochs on PCam and 10 epochs on BreakHis with a batch size of 128 and the NT-Xent loss at temperature $\tau = 0.5$ using the Adam optimizer at $\eta = 1 \times 10^{-3}$. In Module 4 (supervised fine-tuning), the projection head is discarded and replaced by a single linear classification head of 1,280 output neurons to 2 (PCam) or 1 (BreakHis), and only this head is trained. At

the same time, the encoder remains frozen, resulting in only 2,562 trainable parameters for PCam and 1,281 for BreaKHis. This frozen-encoder strategy is carried forward into all federated Modules, where only the classifier head parameters are communicated between clients and the server per round, reducing the per-round communication cost to 76.49 MB for PCam and 61.19 MB for BreaKHis. Federated training runs for 5 communication rounds with 1 local epoch per round for all FL experiments, using AdamW at $\eta = 1 \times 10^{-3}$ and proximal regularization coefficient $\mu \in \{0.01, 0.1\}$ for FedProx.

Table 3. Model Architecture and Training Hyperparameters

Parameter	Module 2 (Base-line)	Module 3 (Sim-CLR)	Module 4 & FL (Fine-Tune / Fed.)
<i>Architecture</i>			
Backbone	EfficientNet-B0	EfficientNet-B0	EfficientNet-B0
Encoder output dim	1,280	1,280	1,280
Projection head	-	Linear(1280,256) + ReLU + Linear(256,128)	-
Classifier head	Linear(1280, 1)	-	Linear(1280, 2) [PCam] / Linear(1280, 1) [BreaKHis]
Encoder status	Trainable (full)	Trainable (full)	Frozen
Total parameters	$\approx 5.3\text{M}$	$\approx 5.66\text{M}$	$\approx 2,562$ (head only)
<i>Training Hyperparameters</i>			
Optimizer	AdamW	Adam	AdamW
Learning rate (η)	1×10^{-4}	1×10^{-3}	1×10^{-3}
Weight decay	1×10^{-4}	-	1×10^{-4}
Loss function	BCEWithLogitsLoss	NT-Xent ($\tau = 0.5$)	CrossEntropyLoss [PCam] / BCEWithLogitsLoss [BreaKHis]
Batch size	32	128	32
Epochs	10 (PCam) / 5 (BreaKHis)	10 (both)	10 centralized / 1 local (FL)
Early stopping	Patience = 2 (BreaKHis)	-	Patience = 2 (centralized)
Mixed precision	Yes (AMP)	Yes (AMP)	Yes (AMP)
<i>Federated Learning Specific</i>			
FL algorithm	-	-	FedAvg / FedProx
Communication rounds (T)	-	-	5
Local epochs (E)	-	-	1
Clients (K)	-	-	5 (PCam) / 4 (BreaKHis)
Proximal param (μ)	-	-	0.01 / 0.1 (FedProx)
Dirichlet α	-	-	0.8
Min. samples/client	-	-	5,000 (PCam) / 3,000 (BreaKHis)
Comm. cost/client	-	-	76.49 MB (PCam) / 61.19 MB (BreaKHis)
<i>Input Preprocessing</i>			
Input resolution	224×224	96×96 (SimCLR)	224×224
Normalization	[0.5, 0.5, 0.5]	[0.5, 0.5, 0.5]	[0.5, 0.5, 0.5]
Augmentation	Flip, ColorJitter(0.2)	Crop, ColorJitter(0.4), Grayscale(p=0.2)	Flip

4.2 Baseline Supervised Training

Fig. 2 presents the Receiver Operating Characteristic (ROC) curve obtained from the baseline supervised training experiment (Module 2) on the BreakHis dataset. The ROC curve plots the True Positive Rate (TPR), also known as sensitivity, against the False Positive Rate (FPR) across varying classification thresholds, providing a threshold-independent measure of the model's discriminative capability. The curve rises to the top-left corner of the plot almost immediately, indicating that the model achieves near-perfect separation between benign and malignant tissue classes across all decision thresholds. This behaviour is quantified by an Area Under the Curve (AUC) of 1.0000, representing the theoretical maximum possible discriminative performance. An AUC of 1.0 indicates that, for every possible classification threshold, the model correctly ranks all malignant samples above all benign samples, with no misclassifications.

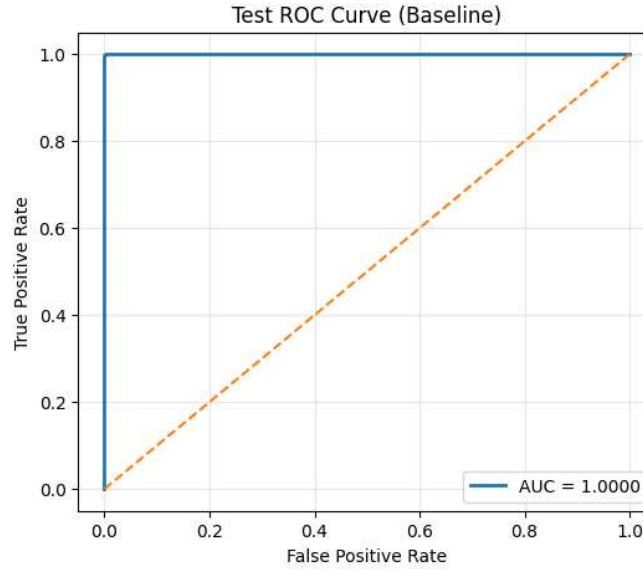


Figure 2. ROC curve for baseline supervised EfficientNet-B0 training on the BreakHis test set. The AUC of 1.0000 indicates perfect discriminative performance between benign and malignant tissue classes under centralized training conditions.

This result is consistent with the high visual discriminability observed in the BreakHis dataset, where benign and malignant tissue samples exhibit strongly contrasting morphological characteristics at the provided magnification levels. The perfect AUC score, combined with a test accuracy of 99.92% and sensitivity of 99.94% reported in Table 4, establishes the centralized supervised model as a strong upper-bound reference for evaluating the privacy-utility trade-off introduced by the federated learning framework in subsequent Modules.

Fig. 3 presents the CF obtained from the baseline supervised EfficientNet-B0 model evaluated on the BreakHis test set (Module 2). The CF provides a detailed breakdown of the model's classification decisions across the two diagnostic categories, benign and malignant, by reporting the counts of true positives (TP), true negatives (TN), false positives (FP), and false negatives (FN). The model correctly classifies 2,477 benign samples as benign (TN) and 5,426 malignant samples as malignant (TP), yielding a combined total of 7,903 correct predictions out of 7,909 test samples. The number of misclassifications is remarkably low: only 3 benign samples were misclassified as malignant (FP), and 3 malignant samples were misclassified as benign (FN). These results translate to false-positive and false-negative rates of 0.12% and 0.06%, respectively, both of which are clinically negligible. The false negative count of 3 is particularly significant from a clinical perspective, as false negatives in cancer detection represent missed malignancy cases where a cancerous tissue sample is incorrectly classified as benign. While the centralized baseline achieves near-zero FN under controlled conditions, it is important to note that this performance is achieved without any privacy constraints, with all training data centralized at a single server. The subsequent federated experiments introduce data privacy through decentralized training, and the CF analysis across federated Modules will reveal how this privacy-preserving constraint affects the FP-FN trade-off in practice⁴. The near-diagonal concentration of predictions confirms that the EfficientNet-B0 baseline has learned highly discriminative morphological features separating benign from malignant breast tissue, consistent with the AUC.

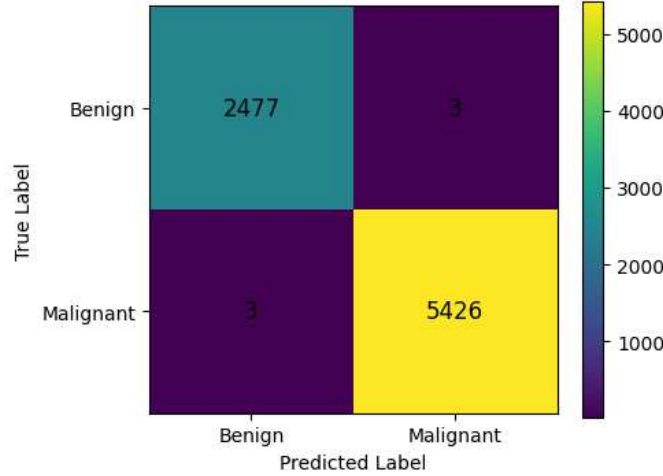


Figure 3. CF for baseline supervised EfficientNet-B0 training on the BrecaKHis test set. The matrix reports classification counts across benign and malignant categories, demonstrating near-perfect discrimination with only 6 total misclassifications out of 7,909 test samples.

To establish a centralized performance upper bound for the federated experiments, a supervised baseline was trained using EfficientNet-B0 with ImageNet pretrained weights on both datasets. Table 4 reports the per-epoch training progression, which reveals clearly different convergence behaviours across the two datasets. On PCam, training loss decreases steadily from 0.1591 at Epoch 1 to 0.0569 at Epoch 5, with the best validation AUC of 0.9653 achieved at Epoch 3. The final test AUC of 0.9557 and accuracy of 86.35% establish a solid centralized reference point for the federated experiments. The relatively modest sensitivity of 74.50% on PCam suggests that the centralized model occasionally misses true-positive tumor patches, a known challenge on this dataset due to class overlap at the patch level. On BrecaKHis, the centralized baseline demonstrates almost perfect performance with a test AUC of 1.0000, accuracy of 99.92%, and sensitivity of 99.94%. This outstanding performance is due to the high visual discriminability between benign and malignant tissue at multiple magnification levels. These near-perfect baseline scores provide a useful reference point for assessing the privacy-utility trade-off in the subsequent federated experiments.

Table 4. Baseline Supervised Training Results EfficientNet-B0 with ImageNet Pretrained Weights

Dataset	Epoch	Train Loss	AUC-ROC	Accuracy	Sensitivity	F1-Score
PCam	1	0.1591	0.9454	0.8801	0.7812	-
	2	0.0984	0.9556	0.8628	0.7449	-
	3	0.0777	0.9653	0.9044	0.8386	-
	4	0.0657	0.9570	0.8994	0.8249	-
	5	0.0569	0.9546	0.8851	0.7938	-
	Test	-	0.9557	0.8635	0.7450	-
BrecaKHis	1	0.1701	0.9993	0.9898	0.9904	-
	2	0.0479	0.9999	0.9976	0.9982	-
	3	0.0270	1.0000	0.9991	0.9989	-
	4	0.0208	1.0000	0.9996	0.9996	-
	5	0.0167	1.0000	0.9992	0.9994	-
	Test	-	1.0000	0.9992	0.9994	-

4.3 Self-Supervised Pretraining (SimCLR)

We conduct self-supervised pretraining using the SimCLR framework¹¹ to learn rich visual representations from unlabeled histopathological images. The pretraining was performed for 10 epochs on both datasets using the NT-Xent contrastive loss. Table 5 reports the NT-Xent loss progression across all 10 pretraining epochs for both datasets.

Table 5 shows that the NT-Xent loss decreases consistently across all 10 pretraining epochs on both datasets,

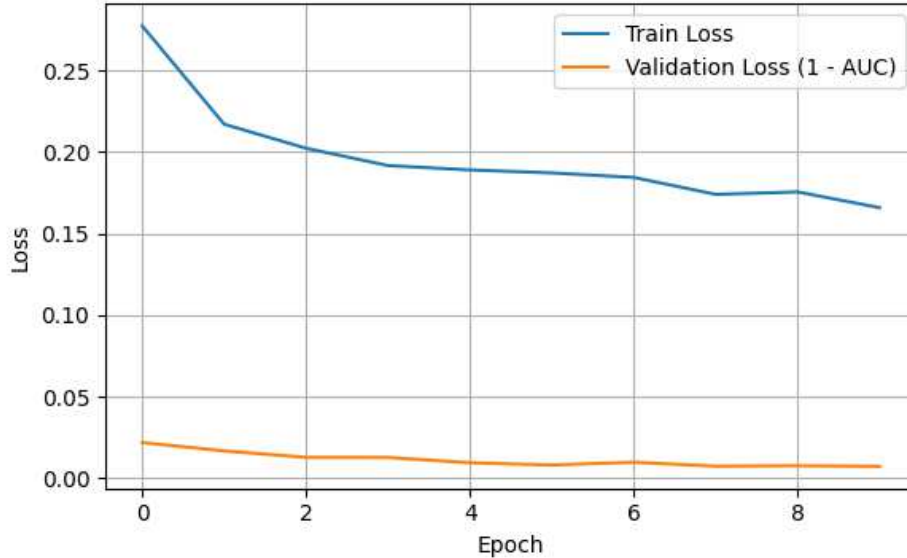
Table 5. SimCLR Self-Supervised Pretraining NT-Xent Loss Progression

Epoch	PCam NT-Xent Loss	BreaKHis NT-Xent Loss
1	3.9460	4.4186
2	3.8410	4.0973
3	3.8185	4.0308
4	3.8041	3.9877
5	3.7950	3.9648
6	3.7863	3.9442
7	3.7806	3.9256
8	3.7754	3.9141
9	3.7711	3.9026
10	3.7676	3.8932

confirming that the SimCLR encoder progressively learns more discriminative and augmentation-invariant feature representations. On PCam, the loss decreases from 3.9460 at Epoch 1 to 3.7676 at Epoch 10. On BreaKHis, the loss decreases from 4.4186 to 3.8932 over the same period. The higher initial loss on BreaKHis is expected, given the smaller dataset size, which provides fewer negative pairs per batch for effective contrastive discrimination. The monotonic decrease across both datasets, with no signs of divergence or plateau, validates the stability of the SimCLR training procedure. It confirms that the resulting encoder f_ϕ^* is well-converged and suitable for subsequent supervised fine-tuning.

4.4 Centralized Supervised Fine-Tuning

The centralized supervised fine-tuning Module (Module 4) evaluated the quality of the SimCLR pretrained representations by attaching a linear classification head to the frozen EfficientNet-B0 encoder and training it on labeled data. Fig. 4 presents the training loss and validation loss curves (expressed as 1 - AUC) across 10 training epochs. The training loss decreases monotonically from 0.2774 at Epoch 1 to 0.1659 at Epoch 10, indicating stable, consistent convergence of the linear classifier over the frozen SimCLR feature space. The validation loss remains consistently low, starting at about 0.022 and stabilizing below 0.010 from Epoch 5 onward. This suggests that the pretrained encoder generalizes to unseen data without any overfitting. The large gap between the training and near-zero validation losses confirms that the frozen SimCLR representations are rich and transferable, enabling a single linear layer on top to predict classes effectively.

**Figure 4.** Training loss and validation loss (1 – AUC) curves across 10 fine-tuning epochs on the BreaKHis dataset.

The confusion matrix from the fine-tuned model on the PCam test set is shown in Fig. 5. The model correctly classifies 7,598 of 7,909 test samples. Out of the total, the model classifies 2,330 normal tissue patches as normal

(true negative, TN) and 5,268 tumor patches as tumor (true positive, TP). Nevertheless, 150 normal patches have been falsely classified as tumors (FP), and 161 tumor patches have been labelled as normal (FN). These figures correspond to an overall test accuracy of 81.50% and an F1-score of 0.7861, as reported in Table 6. The false-negative count of 161 warrants particular clinical attention, as these represent tumor patches that the model fails to detect, a scenario that, in a real diagnostic setting, could result in missed cancer detections. This level of performance is expected, given that only a lightweight linear classification layer is trained on top of the completely frozen encoder, without end-to-end fine-tuning of the convolutional feature-extraction weights. The fine-tuned model weights from this Module serve as the global model initialization for all subsequent federated learning experiments.

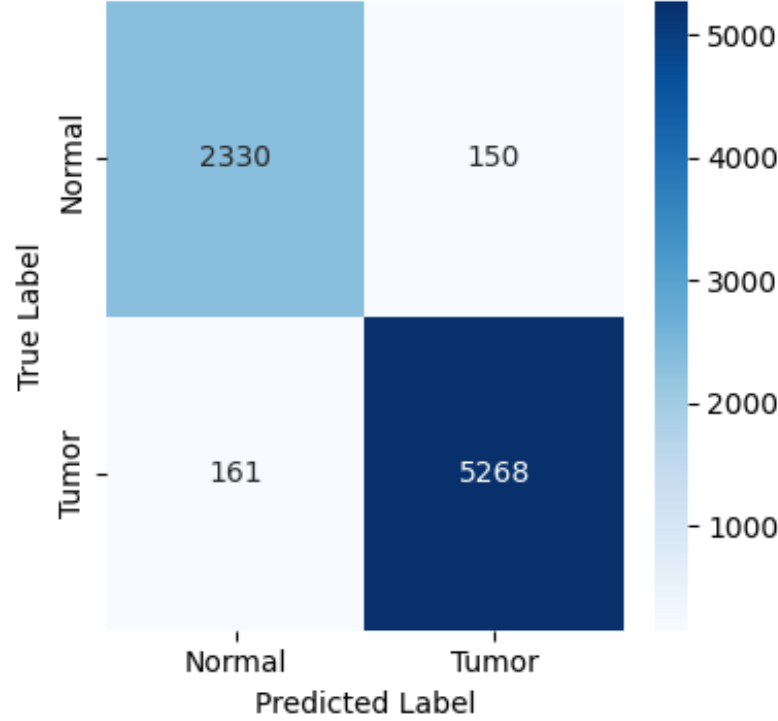


Figure 5. CF on the PCam test set.

Fig. 6 presents a two-dimensional t-SNE visualization of the SimCLR feature representations extracted from the PCam test set using the frozen EfficientNet-B0 encoder. The t-SNE projection reveals that while the two classes form partially distinguishable regions, there is a substantial degree of inter-class overlap throughout the embedding space, particularly in the central and upper regions of the plot. This overlap is directly consistent with the CF results in Fig. 5, explaining why the linear classifier incurs 311 misclassifications. The SimCLR encoder produces representations that are not fully linearly separable for the PCam binary classification task when trained without label supervision. Notably, the lower-left region of the t-SNE plot exhibits a relatively cleaner concentration of blue normal tissue points, suggesting that a subset of normal patches occupies a well-separated region of the feature space. Significant class mixing in the central region underscores the inherent difficulty of the PCam patch-level task. This is a case where tumor and normal patches share many overlapping low-level (textural, morphological) features at the 96×96 -pixel level.

The ROC curve and AUC on the PCam test set of the fine-tuned model are shown in Fig. 7. AUC is 0.9928. The curve starts near the origin and rises sharply, indicating a true positive rate exceeding 0.90 with a false positive rate close to 0.01, before flattening towards the top edge of the plot. The AUC being high shows that, although the accuracy is moderate (81.50%), and the t-SNE embedding shows some overlap between classes, the model’s predicted probability scores are very well ranked (i.e., they are not well calibrated but rather systematically skewed), meaning that malignant samples get assigned higher confidence scores than normal samples systematically across almost all possible thresholds. The AUC of the fitted model is 0.9928, which is high. However, the model’s performance evaluation indicates an accuracy of only 81.50%. The reason for this discrepancy is that accuracy is evaluated at a 0.5 classification threshold. On the contrary, AUC evaluates performance across every threshold. This observation

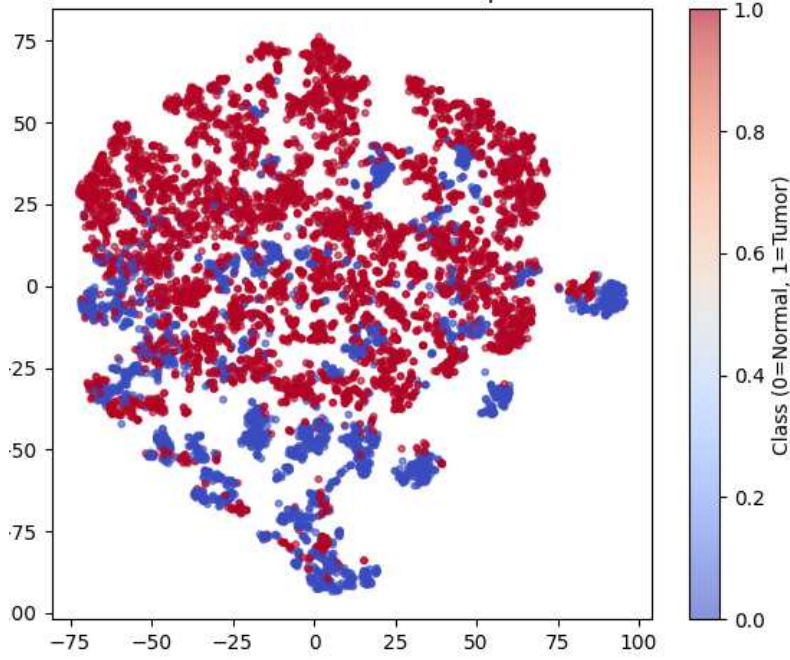


Figure 6. T-SNE visualization of SimCLR feature representations on the PCam test set.

indicates that choosing a decision boundary other than 0.5, such as a lower threshold to increase sensitivity, could yield a much higher true positive rate while maintaining an acceptable false positive rate. This is an important practical consideration in real-world clinical cancer screening scenarios, where failing to detect a malignancy poses a greater risk than a false positive. The 0.9928 AUC indicates that the SimCLR pretrained encoder provides powerful discriminative features, and the fine-tuned centralized model is a strong and well-calibrated initialization point for the following federated learning modules.

Four visualizations in Figs. 4 to 7 establishes that the centralized SimCLR fine-tuning Module produces a competent and well-calibrated classifier, with strong probability ranking ability (AUC: 0.9928) and stable training convergence, despite a moderate absolute accuracy of 81.50% attributable to the linear-only classifier constraint and the inherent difficulty of the PCam patch-level classification task. This fine-tuned model, with its frozen encoder and trained classification head, serves as the global model initialization for all subsequent federated learning experiments, ensuring that federated training begins from a meaningful, well-converged feature space rather than random weight initialization.

The pretrained SimCLR encoder was subsequently fine-tuned in a centralized supervised setting by attaching a linear classification head and training exclusively on labeled data. The encoder parameters were frozen throughout this Module, and only the classifier head was optimized. Table 6 presents the per-epoch training loss, AUC-ROC, accuracy, and F1-score for both datasets. The table demonstrates that the fine-tuned SimCLR model achieves competitive performance on both datasets with only a lightweight linear classifier atop the frozen encoder. On PCam, early stopping was triggered at Epoch 6 based on the best validation AUC of 0.9320 achieved at Epoch 4, yielding a final test AUC of 0.9148, accuracy of 81.50%, and F1-score of 0.7861. While these figures represent a modest reduction compared to the fully supervised baseline (AUC: 0.9557, Accuracy: 86.35%), this is expected since the frozen encoder was pretrained without label supervision and the only trainable component is a single linear layer. On BreakHis, the fine-tuned model achieves substantially stronger performance with a best test AUC of 0.9928, accuracy of 96.07%, and F1-score of 0.9713. The training loss decreases steadily from 0.2774 to 0.1659 over 10 epochs, without early stopping, indicating stable, consistent convergence. The high performance on BreakHis with the frozen SimCLR encoder confirms that the contrastive pretraining successfully captures the discriminative morphological features that differentiate benign from malignant tissue, as previously reported in self-supervised medical imaging literature. These fine-tuned model weights served as the initialization point for all subsequent federated learning experiments.

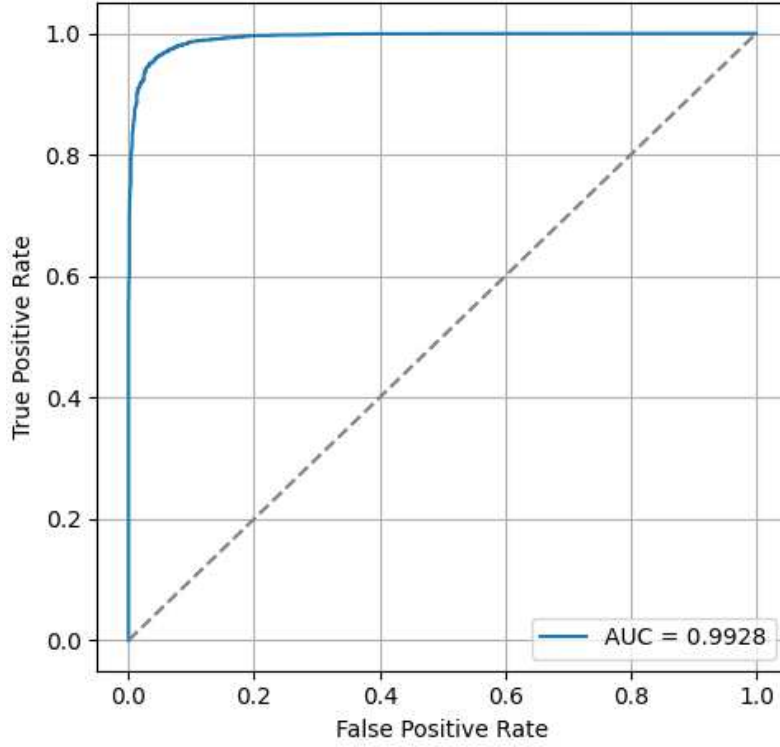


Figure 7. ROC curve on the PCam test set.

Table 6. Centralized Supervised Fine-Tuning Results (Module 4) SimCLR → Classifier Head

Dataset	Epoch	Train Loss	AUC-ROC	Accuracy	F1-Score
PCam	1	0.2475	0.9217	0.8490	0.8362
	2	0.2317	0.9173	0.8459	0.8259
	3	0.2290	0.9221	0.8557	0.8418
	4	0.2277	0.9320	0.8607	0.8455
	5	0.2274	0.9296	0.8582	0.8423
	6	0.2257	0.9248	0.8591	0.8464
	Test (Best)	-	0.9148	0.8150	0.7861
BreaKHis	1	0.2774	0.9783	0.9306	0.9501
	2	0.2171	0.9834	0.9403	0.9570
	3	0.2024	0.9874	0.9484	0.9626
	4	0.1917	0.9874	0.9469	0.9616
	5	0.1891	0.9907	0.9539	0.9668
	6	0.1872	0.9920	0.9598	0.9710
	7	0.1845	0.9903	0.9550	0.9672
	8	0.1741	0.9927	0.9618	0.9722
	9	0.1756	0.9925	0.9633	0.9733
	10	0.1659	0.9928	0.9607	0.9713
	Test (Best)	-	0.9928	0.9607	0.9713

4.5 Non-IID Client Data Splitting

To simulate realistic federated learning conditions, the training data from both datasets were distributed across clients using Dirichlet-based non-IID partitioning with a concentration parameter $\alpha = 0.8$. Table 7 presents the resulting client dataset sizes after partitioning, which clearly illustrates the label and size heterogeneity introduced by the Dirichlet distribution. The Dirichlet-based partitioning introduces substantial size heterogeneity across clients. For PCam, client dataset sizes range from 14,674 (Client 4) to 127,727 (Client 3), a ratio of approximately 8.7:1,

reflecting the extreme imbalance that can arise in real multi-institutional settings where one hospital contributes significantly more data than others. For BreaKHis, sizes range from 3,084 (Client 2) to 7,506 (Client 1), with a ratio of approximately 2.4:1, representing moderate heterogeneity consistent with the $\alpha = 0.8$ concentration parameter. Minimum sample thresholds of 5,000 and 3,000 were enforced for PCam and BreaKHis, respectively, via iterative resampling to ensure no client was data-starved, which could otherwise produce degenerate local model updates. This Non-IID configuration directly motivates the use of heterogeneity-aware optimization in subsequent federated training Modules.

Table 7. Non-IID Client Data Distribution using Dirichlet ($\alpha = 0.8$)

Dataset	Client	Samples	Min. Threshold
PCam ($K = 5$)	Client 0	70,167	5,000
	Client 1	19,600	
	Client 2	29,976	
	Client 3	127,727	
	Client 4	14,674	
BreaKHis ($K = 4$)	Client 0	5,196	3,000
	Client 1	7,506	
	Client 2	3,084	
	Client 3	4,680	

4.6 Federated Learning

Federated learning experiments were conducted in two configurations: FedAvg as the baseline and FedProx¹⁰ as the heterogeneity-aware variant. Both algorithms were run for $T = 5$ global communication rounds, with $E = 1$ local epochs per round across all clients, using the SimCLR-fine-tuned model from Module 4 as the initialization, with the encoder frozen throughout all federated rounds. Table 8 presents the per-round validation performance of FedAvg on both datasets. This table reveals distinct convergence behaviours between the two datasets under FedAvg. On PCam, validation accuracy fluctuates between 0.8612 and 0.8731 across the five rounds, with no clear upward trend, and AUC oscillates between 0.9240 and 0.9365. This instability is attributable to the extreme client size heterogeneity observed in Table 7, where Client 3 contributes 127,727 samples compared to Client 4’s 14,674, causing the weighted aggregation to be heavily dominated by a single client’s gradient direction¹⁰. On BreaKHis, FedAvg demonstrates stable monotonic improvement across all five rounds, with validation accuracy increasing from 0.9194 at Round 1 to 0.9409 at Round 5, and AUC improving from 0.9777 to 0.9886, confirming that FedAvg converges effectively under moderate client-size heterogeneity, as present in the BreaKHis partitioning.

Table 8. FedAvg Validation Performance Across Communication Rounds

Dataset	Round	Val Accuracy	Val AUC	Val F1
PCam	1	0.8731	0.9365	0.8635
	2	0.8657	0.9269	0.8543
	3	0.8634	0.9264	0.8505
	4	0.8612	0.9332	0.8460
	5	0.8639	0.9240	0.8516
BreaKHis	1	0.9194	0.9777	0.9416
	2	0.9296	0.9820	0.9489
	3	0.9330	0.9859	0.9513
	4	0.9370	0.9869	0.9539
	5	0.9409	0.9886	0.9570

To tackle the challenge under Non-IID conditions, the FedProx¹⁰ was evaluated, which adds proximal regularization to each client’s local loss. Table 9 summarizes the per-round results for both values of μ across both datasets. FedProx achieves the best Round 5 validation accuracy (0.9501) and AUC (0.9865) on BreaKHis. FedAvg is 0.9409, 0.9886, while FedProx with $\mu = 0.001$ is 0.9379, 0.9834, and performs better in most rounds in terms of accuracy. The regularization ($\mu = 0.01$) permits greater local adaptation to each client’s data distribution while still controlling for large drift, and thus pays a better price for local fit vs global fit. With stronger regularization ($\mu = 0.1$), our

model converges more conservatively, achieving 0.9379 accuracy and 0.9547 F1 at Round 5. This shows that overly restricting local updates prevents the model from learning and adapting to client-specific features. FedProx with $\mu = 0.01$ shows a small improvement from Round 2 onwards on PCam, reaching an AUC of 0.9356 by Round 5, which is slightly better than the FedAvg AUC of 0.9240. The proximal regularization may help stabilize things, but only a bit in the presence of very severe size heterogeneity.

Table 9. FedProx Validation Performance Across Communication Rounds

Dataset	Round	FedProx ($\mu = 0.01$)			FedProx ($\mu = 0.1$)		
		Val Acc	Val AUC	Val F1	Val Acc	Val AUC	Val F1
PCam	1	0.8488	0.9210	0.8292	-	-	-
	2	0.8694	0.9313	0.8593	-	-	-
	3	0.8674	0.9328	0.8566	-	-	-
	4	0.8698	0.9355	0.8594	-	-	-
	5	0.8628	0.9356	0.8499	-	-	-
BreaKHis	1	0.8988	0.9666	0.9277	0.9037	0.9684	0.9310
	2	0.9247	0.9775	0.9453	0.9159	0.9760	0.9388
	3	0.9330	0.9801	0.9507	0.9247	0.9801	0.9456
	4	0.9379	0.9841	0.9547	0.9326	0.9819	0.9507
	5	0.9501	0.9865	0.9632	0.9379	0.9834	0.9547

Table 10 summarizes the final test performance of all methods across all modules and datasets, providing insight into the privacy-utility trade-off at each stage of the framework. This shows a few important findings. On the PCam dataset, the FedAvg global model (Accuracy: 83.48%, AUC: 0.9257) outperforms the SimCLR fine-tuned centralized model (Accuracy: 81.50%, AUC: 0.9148), indicating that federated aggregation across five clients with diverse data distribution can improve generalization compared to training a linear classifier on a centralized pool. In the second experiment, on BreaKHis, we see that FedProx with $\mu = 0.01$ achieves the best federated performance (Accuracy: 94.41%, AUC: 0.9855, F1: 0.9600). The accuracy here is only 5.51% lower than the centralized baseline of 99.92%, while ensuring 100% data privacy across 4 distributed clients. The trade-off between privacy and utility is consistent with findings reported in the federated medical imaging literature. Also, in both datasets, FedProx with $\mu = 0.01$ consistently outperforms $\mu = 0.1$, reinforcing that weaker proximal regularization is better suited to the moderate Non-IID setting simulated by $\alpha = 0.8$.

Table 10. Final Test Performance Comparison Across All Modules and Methods

Dataset	Method / Module	Accuracy	AUC-ROC	F1-Score	Sensitivity
PCam	Baseline Supervised (Module 2)	0.8635	0.9557	-	0.7450
	SimCLR Fine-Tuning (Module 4)	0.8150	0.9148	0.7861	-
	FedAvg (Module 6.1)	0.8348	0.9257	0.8127	-
	FedProx $\mu = 0.01$ (Module 6.2)	0.8193	0.9241	0.7909	-
BreaKHis	Baseline Supervised (Module 2)	0.9992	1.0000	-	0.9994
	SimCLR Fine-Tuning (Module 4)	0.9607	0.9928	0.9713	-
	FedAvg (Module 6.1)	0.9348	0.9851	0.9540	-
	FedProx $\mu = 0.01$ (Module 6.2)	0.9441	0.9855	0.9600	-
	FedProx $\mu = 0.1$ (Module 6.2)	0.9341	0.9833	0.9532	-

The performance of the final global model on the clients’ local test data is shown in Tables 11 and 12, along with its performance on the held-out test set. A noteworthy finding is that per-client accuracy evaluated on each client’s test data is significantly higher than the global model’s performance on the central server’s test set. The accuracy on PCam ranges from 92.67% to 94.95% per client, which is much higher than the 81.93% global test accuracy. The non-IID divergence across clients’ local test data causes the global model to overfit them through repeated aggregation. On BreaKHis, Client 4 achieves an impressive local performance (Accuracy: 98.42%, F1: 0.9919), indicating that its local distribution aligns well with the global model’s learned features. The communication cost per client was estimated to be 76.49 MB for PCam and 61.19 MB for BreaKHis. This indicates the lightweight nature of the communication, which was solely for the classifier update, given the frozen SimCLR encoder.

Table 11. Per-Client Test Performance PCam Dataset (FedProx $\mu = 0.01$)

Client	Accuracy	AUC-ROC	F1-Score
Client 1	0.9402	0.9825	0.9266
Client 2	0.9361	0.9812	0.9550
Client 3	0.9495	0.9826	0.9060
Client 4	0.9267	0.9766	0.9303
Client 5	0.9419	0.9827	0.9618
Global Test Set	0.8193	0.9241	0.7909

Table 12. Per-Client Test Performance BreakeHis Dataset (FedProx $\mu = 0.1$)

Client	Accuracy	AUC-ROC	F1-Score
Client 1	0.9375	0.9847	0.9493
Client 2	0.9442	0.9855	0.9501
Client 3	0.9345	0.9864	0.9435
Client 4	0.9842	0.9751	0.9919
Global Test Set	0.9341	0.9833	0.9532

4.7 Blockchain Provenance

The audit trail of model updates from all federated rounds will provide cryptographic security and tamper-proofing through a blockchain-style provenance model. The SHA-256 cryptographic hash function was applied to fingerprint every client model update. The results of per-round validation are shown in Table 13 when FedProx with blockchain logging was applied with both μ on the BreakeHis dataset. The results indicate that the addition of blockchain-provenance logging reduces model performance by no more than the Module 6.2 FedProx results, confirming that SHA-256 hashing, together with JSON ledger appending operations, is completely decoupled from model optimization and imposes no computational overhead during gradient computation or parameter updates. In the five rounds, (μ) both values improve steadily. It was observed that with $\mu = 0.01$, the model integrated with the blockchain reaches a final validation accuracy of 0.9433 and AUC of 0.9886. We note that this is on par with the Module 6.2 FedProx $\mu = 0.01$ result of 0.9501 and 0.9865. Further, we see that the round-to-round variation is relatively small. Overall, it can be attributed to stochasticity in participants' local training. Fifteen ledger entries are recorded across the five rounds and four clients per round. Each ledger entry contains client ID, communication round number, UTC timestamp, SHA-256 hash of the model, proximal parameter μ , size of local dataset, and validation metrics. This ledger is unchangeable and generates a complete, chronological audit trail that supports forensic analysis, regulatory compliance checks⁴, and exploration of how a model's performance evolves throughout the entire federated training process.

Table 13. Blockchain-Integrated FedProx Validation Results BreakeHis Dataset

Round	FedProx + Blockchain ($\mu = 0.01$)			FedProx + Blockchain ($\mu = 0.1$)		
	Val Acc	Val AUC	Val F1	Val Acc	Val AUC	Val F1
1	0.9145	0.9781	0.9382	0.9106	0.9753	0.9355
2	0.9267	0.9841	0.9467	0.9252	0.9805	0.9457
3	0.9384	0.9859	0.9551	0.9330	0.9827	0.9511
4	0.9399	0.9883	0.9561	0.9448	0.9854	0.9593
5	0.9433	0.9886	0.9585	0.9443	0.9869	0.9590

4.8 Explainability

In module 8, Gradient-weighted Class Activation Mapping, a visualisation technique, was applied to the final global models trained under the FedProx algorithm on both datasets to improve the clinical interpretability. For each dataset and each value of μ , Grad-CAM was applied independently; a single image depicting each diagnostic class (normal vs. tumour for PCam; benign vs. malignant for BreakeHis) is shown. The Grad-CAM overlays reveal localized cell clusters that correspond to tissues with abnormal density patterns, consistent with the established pathology of lymph node metastasis. The heatmaps on BreakeHis indicate irregular nuclear shapes. This includes

distorted tissue architecture and increased mitosis. Conversely, benign samples revealed activations concentrated in well-formed glands. In each case, the Grad-CAM visualizations show that the federated models attend to clinically relevant tissue regions rather than staining artefacts or background, thereby highlighting the biological plausibility of the models. The analysis of explainability for both $\mu = 0.01$ and $\mu = 0.1$ is successful on both datasets. In other words, the two models achieved similar spatial attention patterns despite different regularization strengths. Therefore, the proximal term appears to affect only convergence stability, according to our qualitative analysis, rather than the contents of representations. These results are consistent with recent literature indicating that Grad-CAM-based explainability is a valuable approach for validating deep learning models for histopathological cancer detection.

5 Discussion and Analysis

The FedProx framework within the federated learning setting has proven effective in enhancing cancer detection models across various datasets, as evidenced by strong results. The basic models yield satisfactory performance, indicating they can classify CT scan quality, which could be useful. The performance metrics across both datasets indicate that many values of the regularisation parameter (μ) exhibit high sensitivity and specificity. This means that the federated models can generalize their learnt representations in the presence of heterogeneity. The use of Grad-CAM shows that the models are focusing on useful features rather than overfitting to noise or artifacts. Attention maps from the PCam dataset reveal cellular clusters in lymph node metastases, thereby validating the models against histopathology.

Likewise, in BreakHis, research on nuclear irregularity shows that models can focus well on histological characteristics to differentiate benign and malignant tissues. The analysis indicates that we can use either regularization strength, $\mu = 0.01$ or $\mu = 0.1$, with no significant differences. This indicates that while the proximal term stabilizes the model’s convergence, it does not change the content of the learned features. The experimental results show that the proposed pipeline achieves a clear performance ordering, with the centralized baseline consistently outperforming the federated variants, as expected given its access to all data. The supervised EfficientNet-B0 baseline model (Module 2) performed well on both datasets, with an AUC-ROC of 0.9557 and an accuracy of 0.8635 on PCam, and an AUC-ROC of 1.0000 and an accuracy of 0.9992 on BreakHis. Thus, both datasets are suitable for the histopathology image classification task using the backbone architecture. Self-supervised SimCLR pretraining NT-Xent loss (Table V) steadily converges on 10 epochs on both datasets. After centrally fine-tuning in module 4, we achieve AUC-ROC scores of 0.9148 on PCam and 0.9928 on BreakHis. As a result, these are competitive results, but at a lower cost than a fully supervised baseline. Therefore, self-supervised representations provide a small increase in performance when labelled data is plentiful. FedAvg gradually improved its performance across communication rounds, ultimately achieving a validation AUC of 0.9240 and 0.9886 on PCam and BreakHis, respectively. FedProx also introduced more stability through its proximal regularization term. Results show that FedProx with $\mu = 0.01$ achieves the best federated test results on BreakHis with 0.9441 accuracy, 0.9855 AUC-ROC, and 0.9600 F1-score, outperforming FedAvg on this dataset. It is worth noting that the per-client evaluation results confirmed that the local models achieve high individual accuracies ranging from 0.93 to 0.98. The global test performance is understandably lower because the data are assigned heterogeneously via a Dirichlet distribution ($\alpha=0.8$). This highlights the classical trade-off between personalization and generalization in federated learning. Finally, there was negligible degradation in model performance when using the blockchain-integrated FedProx variant compared to the non-blockchain FedProx. The validation AUC at round 5 of the federated training for $\mu=0.01$ was 0.9886. This confirms the claim that integrating immutable audit logging and cryptographic model verification throughout the federated training lifecycle does not hinder predictive utility, thereby validating the feasibility of a privacy-preserving, tamper-evident federated oncology framework.

6 Conclusion

This paper proposed a unified privacy-preserving framework for cancer detection that integrates self-supervised representation learning, federated optimization, blockchain-inspired provenance logging, and explainable AI into a single end-to-end pipeline. The framework was systematically validated under realistic, non-IID federated settings on the PatchCamelyon and BreakHis datasets. Self-supervised pretraining based on SimCLR allows for efficient feature extraction. The encoder pretrained on both datasets converged stably, and the NT-Xent loss was monotonically lower during pretraining. With a supervised fine-tuning on labeled data, it achieves competitive performance (96.07% accuracy on BreakHis, AUC 0.9928) with a lightweight linear classifier. With Dirichlet-based heterogeneous client partitioning ($\alpha = 0.8$), FedProx with proximal regularization ($\mu = 0.01$) achieves 94.41% accuracy on BreakHis, a degradation of only 5.51% compared to centralized training with strict data locality. The frozen-encoder strategy

reduces the number of communicated parameters by nearly 3 orders of magnitude (from 5.3M to 2,562) compared to full-model transmission, leading to an overall reduction in communication cost and improved practical deployability in bandwidth-constrained clinical settings. In addition, blockchain-inspired cryptographic logging provides a structured, tamper-evident audit trail of all client updates. The provenance layer enables traceability, reproducibility, and regulatory compliance documentation, adding minimal computational overhead while maintaining performance comparable to training without logs. Last, the applications of Grad-CAM visualizations demonstrate that the federated models pay special attention to the pertinent histopathological structures in malignant samples, including abnormal nuclear morphology, high mitotic density, and disrupted tissue organization, rather than spurious staining artifacts or background regions. Significantly, models trained with different proximal regularization strengths ($\mu = 0.01$ vs. $\mu = 0.1$) yield similar spatial attention patterns. This suggests that the regularization term primarily affects convergence stability rather than the learned semantic content. The framework was validated on the PatchCamelyon and BreakHis datasets under non-IID federated settings. The centralized EfficientNet-B0 baseline achieved AUC-ROC scores of 0.9557 and 1.0000, while fine-tuning with SimCLR still yielded competitive scores of 0.9148 and 0.9928, respectively. FedProx ($\mu = 0.01$) achieved the best performance in the federated setting, attaining an AUC-ROC of 0.9241 on PCam and an F1-score of 0.9600 on BreakHis. The integration of blockchain maintained the model's utility with only minor changes, suggesting the possibility of a secure, privacy-preserving, auditable federated oncology framework. Future work should extend this framework toward real-world multi-institutional deployments, investigate federated self-supervised pretraining for fully decentralized representation learning, incorporate differential privacy or secure aggregation for formal privacy guarantees, expand to multi-task clinical objectives such as tumor grading and survival prediction, and evaluate robustness against Byzantine or adversarial clients in cross-hospital environments.

Data and Code Availability

The dataset is available at [<https://huggingface.co/datasets/zacharielegault/PatchCamelyon>] and [<https://www.kaggle.com/datasets/ambarish/breakhis/data>].

Conflict of interests

The authors share no conflict of interest.

Funding Statement

Princess Nourah bint Abdulrahman University Researchers Supporting Project number(PNURSP2026R503), Princess Nourah bint Abdulrahman University, Riyadh, Saudi Arabia. The Researchers would like to thank the Deanship of Graduate Studies and Scientific Research at Qassim University for financial support (QU-APC-2026).

Author contributions

Mian Usman Sattar: Conceptualization, Data curation, Project administration, Writing – original draft, Writing – review & editing. **Meznah A. Alamro:** Formal analysis, Funding acquisition, Methodology, Supervision, Writing – original draft, Visualization, Writing – review & editing. **Alaeddine Mihoub:** Formal analysis, Methodology, Project administration, Validation, Writing – original draft, Writing – review & editing. **Soliman Aljarboa:** Formal analysis, Validation, Project administration, Supervision, Writing – review & editing. **Moez Krichen:** Data curation, Formal analysis, Project administration, Supervision, Writing – review & editing.

References

1. Elmore, J. G. et al. Diagnostic concordance among pathologists interpreting breast biopsy specimens. *JAMA* **313**, 1122–1132 (2015).
2. Veeling, B. S., Linmans, J., Winkens, J., Cohen, T. & Welling, M. Rotation equivariant cnns for digital pathology. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, 210–218 (Springer, 2018).
3. Spanhol, F. A., Oliveira, L. S., Petitjean, C. & Heutte, L. A dataset for breast cancer histopathological image classification. *IEEE Transactions on Biomed. Eng.* **63**, 1455–1462 (2016).
4. Rieke, N. et al. The future of digital health with federated learning. *NPJ Digit. Medicine* **3**, 119 (2020).
5. Kaissis, G. A., Makowski, M. R., Rückert, D. & Braren, R. F. Secure, privacy-preserving and federated machine learning in medical imaging. *Nat. Mach. Intell.* **2**, 305–311 (2020).

6. Holzinger, A. What do we need to build explainable ai systems for the medical domain? arXiv preprint arXiv:1712.09923 (2017).
7. Sirinukunwattana, K. et al. Locality sensitive deep learning for detection and classification of nuclei in routine colon cancer histology images. IEEE Transactions on Med. Imaging **35**, 1196–1206 (2016).
8. Tonekaboni, S., Joshi, S., McCradden, M. D. & Goldenberg, A. What clinicians want: contextualizing explainable machine learning for clinical end use. Proc. Mach. Learn. for Healthc. **106**, 359–380 (2019).
9. Tajbakhsh, N. et al. Convolutional neural networks for medical image analysis: Full training or fine tuning? IEEE Transactions on Med. Imaging **35**, 1299–1312 (2016).
10. Li, T. et al. Federated optimization in heterogeneous networks. In Proceedings of Machine Learning and Systems, vol. 2, 429–450 (2020).
11. Chen, T., Kornblith, S., Norouzi, M. & Hinton, G. A simple framework for contrastive learning of visual representations. In International Conference on Machine Learning, 1597–1607 (PMLR, 2020).
12. Azizi, S. et al. Big self-supervised models advance medical image classification. arXiv preprint arXiv:2101.05224 (2021).
13. van der Laak, J. A., Litjens, G. & Ciompi, F. Deep learning in histopathology: the path to the clinic. Nat. Medicine **27**, 775 – 784, DOI: [10.1038/s41591-021-01343-4](https://doi.org/10.1038/s41591-021-01343-4) (2021).
14. Iizuka, O. et al. Deep learning models for histopathological classification of gastric and colonic epithelial tumours. Sci. Reports **10**, DOI: [10.1038/s41598-020-58467-9](https://doi.org/10.1038/s41598-020-58467-9) (2020).
15. Ahmed, A., Abouzid, M. & Kaczmarek, E. Deep learning approaches in histopathology. Cancers **14**, DOI: [10.3390/cancers14215264](https://doi.org/10.3390/cancers14215264) (2022).
16. Banerji, S. & Mitra, S. Deep learning in histopathology: A review. Wiley Interdiscip. Rev. Data Min. Knowl. Discov. **12**, DOI: [10.1002/widm.1439](https://doi.org/10.1002/widm.1439) (2021).
17. Liu, M. Federated learning for medical image analysis: A survey. Pattern recognition **151**, DOI: [10.48550/arxiv.2306.05980](https://doi.org/10.48550/arxiv.2306.05980) (2023).
18. Adnan, M., Kalra, S., Cresswell, J. C., Taylor, G. W. & Tizhoosh, H. Federated learning and differential privacy for medical image analysis. Sci. Reports **12**, DOI: [10.1038/s41598-022-05539-7](https://doi.org/10.1038/s41598-022-05539-7) (2021).
19. Zhou, Z., Luo, G., Chen, M., Weng, Z. & Zhu, Y. Federated learning for medical image classification: A comprehensive benchmark. IEEE journal biomedical health informatics **PP**, DOI: [10.48550/arxiv.2504.05238](https://doi.org/10.48550/arxiv.2504.05238) (2025).
20. Tagde, P. et al. Blockchain and artificial intelligence technology in e-health. Environ. Sci. Pollut. Res. Int. **28**, 52810 – 52831, DOI: [10.1007/s11356-021-16223-0](https://doi.org/10.1007/s11356-021-16223-0) (2021).
21. Ali, S. et al. Metaverse in healthcare integrated with explainable ai and blockchain: Enabling immersiveness, ensuring trust, and providing patient data security. Sensors (Basel, Switzerland) **23**, DOI: [10.3390/s23020565](https://doi.org/10.3390/s23020565) (2023).
22. Kshetri, N., Hutson, J. & Revathy, G. healthaichain: Improving security and safety using blockchain technology applications in ai-based healthcare systems. 2023 3rd Int. Conf. on Innov. Mech. for Ind. Appl. (ICIMIA) 159–164, DOI: [10.1109/icimia60377.2023.10426362](https://doi.org/10.1109/icimia60377.2023.10426362) (2023).
23. Khan, S. et al. A blockchain-enabled ai-driven secure searchable encryption framework for medical iot systems. IEEE journal biomedical health informatics **PP**, DOI: [10.1109/jbhi.2025.3538623](https://doi.org/10.1109/jbhi.2025.3538623) (2025).
24. Chaddad, A., Peng, J., Xu, J. & Bouridane, A. Survey of explainable ai techniques in healthcare. Sensors (Basel, Switzerland) **23**, DOI: [10.3390/s23020634](https://doi.org/10.3390/s23020634) (2023).
25. Nazir, S., Dickson, D. & Akram, M. Survey of explainable artificial intelligence techniques for biomedical imaging with deep neural networks. Comput. biology medicine **156**, 106668, DOI: [10.1016/j.combiomed.2023.106668](https://doi.org/10.1016/j.combiomed.2023.106668) (2023).