# Progress Seminar

## Blockchain-Enabled Secure and Transparent Lung Disease Classification in Medical Imaging

*Under the Supervision of*

**Dr. Prabina Pattanayak and Dr. Chandrajit Choudhury**


*Presented By*

**Kirti Swagat Mohanty**

**21-3-04-068**


Photo by Pexels


**Department of Electronics and Communication Engineering,**

**National Institute of Technology, Silchar**

# Table of Contents



# Problem Statement-1- Completed

**Designing a Contrastive Learning Based Robust Multi-label Chest Disease Classification System**

# Problem Statement-2- Implementation Completed

**Blockchain-Secured Federated Learning for Privacy-Preserving Multi-label Chest Disease Classification**

# Problem Statement-3- Yet to Start

**Designing a deep-learning based architecture to optimize features from the input image to enhance the classification**

# Introduction & Background

- The detection and classification of lung diseases from medical images pose significant challenges due to the complexity of pathologies and the need for high diagnosis accuracy.
- However , widespread clinical deployment of AI models is impeded by data sharing barriers arising from **privacy regulations and regulations and institutional silos**, which prevent pooling of diverse, multi-institutional datasets essential for robust model robust model generalization.
- **Model provenance and trust issues**, as centralized repositories lack immutable audit trails to verify that submitted model updates or model updates or weights have not been tampered.
- Block chain  technology addresses these challenges by providing:
    - **Decentralization**: Eliminates single points of failure, enabling hospitals and research institutes to contribute model updates model updates without relinquishing data ownership.
    - **Immutability**: Cryptographic chaining of blocks ensures that once a model hash is recorded, it cannot be altered, enabling altered, enabling verifiable provenance for every model version and audit ready logs of submissions and accesses.
      accesses.
    - **Smart Contracts**: Automate model registration, validation triggers and access control, enforcing predefined rules without rules without interventions.

# Research Motivation and Objectives

## Motivation

- Clinical AI for lung-disease classification demands large, diverse imaging datasets, yet privacy regulations (HIPAA, GDPR) and institutional silos prevent cross-site data sharing.
- Federated learning addresses raw-data privacy but offers no immutable audit trail for model updates, leaving provenance and trust gaps in collaborative AI pipelines.
- Centralized model repositories risk tampering, version drift, and unilateral control—undermining reproducibility and regulatory compliance.
- On-chain storage of full neural networks is infeasible due to block-size limits; off-chain IPFS solutions introduce their own access-control and integrity challenge.

# Research Motivation and Objectives

## Objectives

1. **Design smart contracts that**:
    - Register model submissions on-chain, embedding cryptographic hashes for immutable provenance.
    - Automate performance triggers (e.g., AUC, F1 thresholds) via oracle-fed validation events.
    - Enforce role-based access (contributors, aggregator, end users).
2. **Integrate IPFS for off-chain storage of model weights and explainability artifacts, ensuring:**
    - Deterministic retrieval via content identifiers (CIDs) stored on-chain.
    - Encryption-before-pinning and proxy re-encryption key management for HIPAA-compliant data confidentiality.
3. **Implement multi-layer validation mechanisms encompassing:**
    - Cryptographic integrity checks (SHA-256 hash matching) between on-chain records and IPFS downloads.
    - Performance verification on sealed test sets, recording metrics immutably on blockchain.

# Research Motivation and Objectives

## Objectives

**4. Demonstrate a reference architecture that**:

- Tracks model workflow end-to-end (training → upload → verify → aggregate → deploy).
- Provides audit-ready logs for regulatory and clinical governance.
- Ensures scalability through a hybrid on-chain/off-chain design without sacrificing throughput.

# Covered Objectives

1. **Decentralized Control & Audit of Model Operations**: Design a permissioned blockchain to replace the centralized server, allowing edge nodes to publish encrypted model updates as transactions that are verified and ordered by consensus.

    ✅ Smart Contract Model Registry: *The contract maintains on-chain records for model uploads, including storing the uploader's address, model identifiers, and IPFS hashes. This provides traceability, integrity, and auditability of each model version.*

    ✅ Vendor (Participant) Whitelisting: *Only whitelisted addresses can use certain contract functions, enforcing role-based access and some governance.*

    ✅ No centralized Federated Learning Server: *While not a full decentralized federated learning controller, on-chain storage of model metadata replaces traditional centralized registries in FL.*

    `

2. **Secure Aggregation and Privacy**:

    ✅ No On-chain Data/Update Aggregation: *The smart contract stores model hashes and metadata, but does not aggregate FL updates on-chain nor enforce secure aggregation protocols (e.g., homomorphic encryption, secure enclaves, or additive masking).*

    ✅ Data Privacy: *No raw data or gradients are ever uploaded; the only reference on-chain is the IPFS hash of trained model binaries or snapshots.*

    ✅ Integrity: *The blockchain ensures that each model upload or update can be immutably traced to the uploader.*

# Covered Objectives Contd..

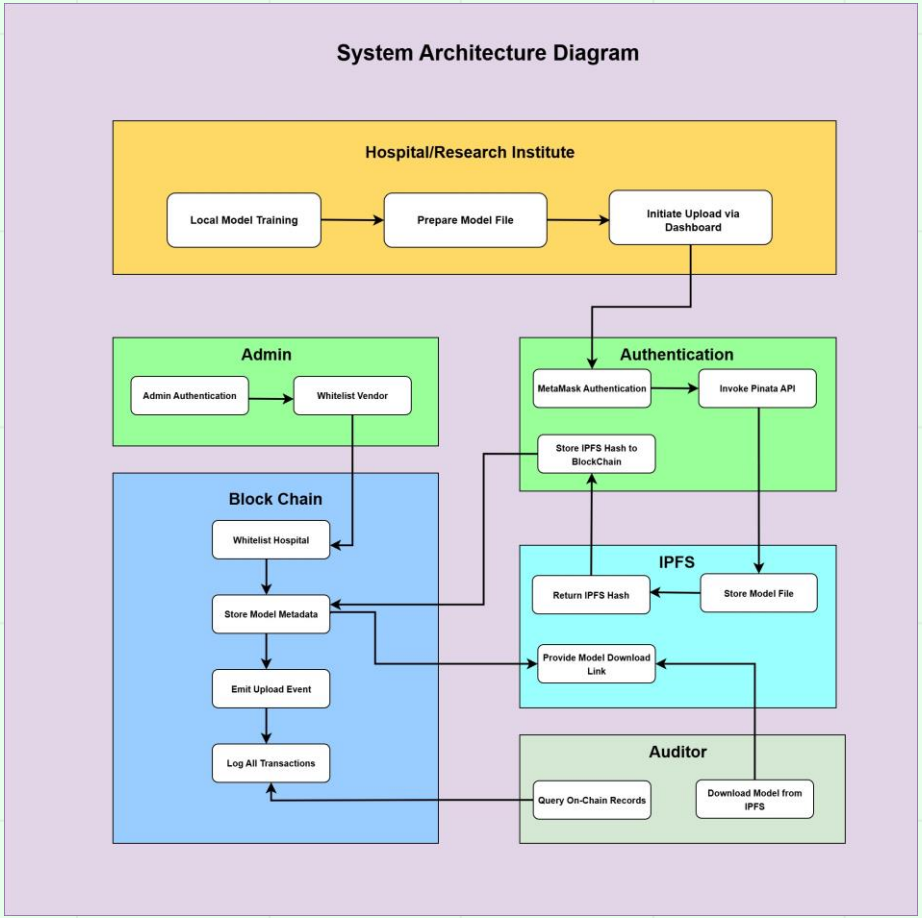
**3. Confidentiality, Integrity, Traceability**:

- ✅ Confidentiality: *Direct model contents are stored on IPFS, not the blockchain itself. Patient data never leaves hospital boundaries, aligning with privacy requirements.*
- ✅ Integrity & Traceability: *Every transaction (upload, whitelist, role assignment) is permanently logged, so model provenance is always auditable to participants.*

**4. Governance Mechanisms**:

- ✅ Whitelisting/Vendor Governance : *Administrators can whitelist or remove vendors on-chain, but there are no tokenomics, rewards, or penalties implemented in the contract*

# System Architecture Overview



# DApps- Decentralized Application

- A **Decentralized Application (DApp)** is an app that interacts with a blockchain through smart contracts, typically offering:
  - A web frontend (often JavaScript frameworks like React)
  - Blockchain-based authentication (wallets such as MetaMask)
  - Direct calls to one or more smart contracts for core functions (no central backend server for main workflows)
  - Use of decentralized storage (e.g., IPFS) for files or metadata

# DApp Components in our work:

| Module | File Name | Reason/Description |
|---|---|---|
| Model Uploader | ModelUploader.js | Uploads ML models to IPFS, interacts with blockchain smart contract, requires MetaMask. |
| Block Chain Connection | BlockchainConnect.js | Handles wallet connections, checks account and network, required for blockchain DApp interactions. |
| Model Registry | AIModelRegistryABI.json | Defines the smart contract interface; the backend logic for decentralized registry and audit. |

# Blockchain Smart Contract

## 1. Whitelist Hospital

- ***Purpose***: Manages approved participant access.
- ***Functionality:***
    - Keeps a mapping (whitelistedVendors[address]) of which hospital/participant addresses are authorized to interact with the contract.
    - Only whitelisted accounts can upload models or perform restricted operations.

## 2. Store Model Metadata

- ***Purpose:*** Registers AI model versions and their storage references.
- ***Functionality:***
    - Accepts a model name and an IPFS hash when a hospital uploads a new model.
    - Records this pairing (storeModel(string modelName, string modelHash)) along with the address of the uploader.
    - Ensures the traceability and transparency of every submitted model.

# Blockchain Smart Contract

### 3. Emit Events

- ***Purpose***: Provides an immutable audit trail and system transparency.
- ***Functionality:***
    - Emits events on critical actions such as:
        - **ModelStored**: Triggered when new model metadata is recorded.
        - **VendorWhitelisted**: Announced whenever a hospital/participant is authorized.
        - **VendorRemoved**: Indicates removal of participant's authorization.

### 4. Log All Transactions

- ***Purpose:*** Ensures transparency, compliance, and accountability..
- ***Functionality:***
    - Every model upload, vendor whitelist/removal, and admin action is written to the blockchain.
    - The log is public and immutable, enabling compliance reviews and dispute resolution.

# How these work together

- **Admin** manages whitelisting and roles.
- **Hospitals (when whitelisted)** can upload models (metadata only), triggering on-chain events.
- **Auditors/External** Users can verify every transaction, download model artifacts by IPFS hash, and confirm participant status—all through the public blockchain.
- These sub-blocks collectively enforce trust, security, and auditability in your decentralized federated learning ecosystem.

| Sub-Block | Blockchain Integration Role |
|---|---|
| Whitelist Hospital | Access control and permission management |
| Store Model Metadata | Decentralized registry of model submissions |
| Emit Events | Transparent audit trail of key actions |
| Role Management | Assignment and enforcement of user/admin vendor rights |
| Log All Transactions | Immutable, public activity and state logging |
| Verification & Querying | Public checks for permissions, models, and role status |

# Auditor

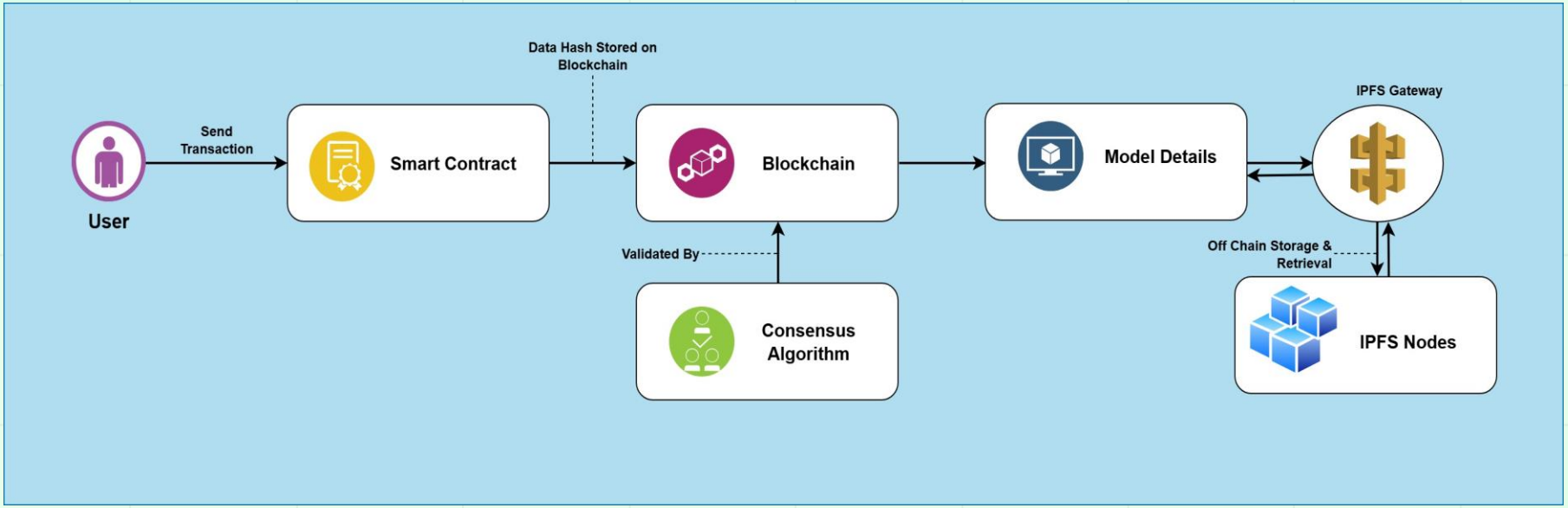
## 1. Query On-Chain Records

- ***Purpose***: Allows auditors and other users to access and review the blockchain ledger to examine every approved action.
- ***Functionality:***
    - Retrieve immutable logs of all model uploads (including model name, IPFS hash, and uploader).
    - Track participant whitelisting, removals, and admin actions through events.
    - Verify timestamps and sequence of transactions for compliance and auditing.

## 2. Download Model from IPFS

- ***Purpose:*** Enables retrieval of the actual machine learning models for evaluation, validation, or research purposes.
- ***Functionality:***
    - Uses publicly stored IPFS hashes (from blockchain registry) to fetch model binaries directly from decentralized storage.
    - Can cross-check that the on-chain hash matches the file content, validating authenticity.

# Blockchain Data Flow



# IPFS – Inter Planetary File System

**It is a peer-to-peer, content-addressed protocol and network designed to create a decentralized, resilient, and verifiable file storage and sharing system.**

**Content Addressing and CIDs**

- Traditional web protocols (HTTP/HTTPS) use location-based addressing, where a URL points to a specific server and path.
- IPFS uses content-based addressing: each file (or data "object") is cryptographically hashed to produce a **Content Identifier (CID).**
- The CID uniquely represents the file's content: any change in the file yields a new CID.
- To retrieve data, peers query the network for the CID instead of a URL.

**Data Chunking and Merkle DAG**

- When a file is added to IPFS:
    - It is split into fixed-size blocks.
    - Each block is hashed, producing its own CID.
    - Blocks are organized into a **Merkle Directed Acyclic Graph (DAG)**, enabling efficient versioning, deduplication, and integrity verification.

# IPFS - InterPlanetary File System

**Distributed Hash Table (DHT)**

- IPFS nodes form a peer-to-peer network. A Distributed Hash Table (DHT) maps CIDs to the addresses of nodes storing the corresponding blocks.
- To fetch a CID, a node locates peers advertising that CID in the DHT, then downloads blocks in parallel from multiple sources.

**Benefits of IPFS**

- *Decentralization & Resilience*: Files are served by any node holding the data; no single point of failure or censorship.
- *Integrity & Immutability*: Content addressing and Merkle DAG ensure tamper resistance—any alteration breaks the CID chain.
- *Versioning*: Merkle DAG inherently tracks historical versions; each update yields a new CID, while preserving links to prior versions.
- *Efficiency*: Parallel block downloads from multiple peers accelerate retrieval, and deduplication conserves bandwidth and storage.

# IPFS in my work

Our work, holds a sophisticated IPFS Integration as follows:

| Step | Process | Technology |
|---|---|---|
| 1. File Upload | User Selects AI Model | React File Input, Form Data |
| 2. IPFS Storage | Upload to Pinata via API | Pinata Cloud API, JWT Authentication |
| 3. Hash Generation | IPFS generates unique CID | Content Addressing (CID) |
| 4. Blockchain Storage | Store model name and IPFS hash | Smart Contract, ether js |
| 5. File Retrieval | Download via IPFS Gateway | Gateway URL with blob handling |

**The IPFS implementation provides:**

- **Decentralized Storage**: Files distributed across IPFS network
- **Content Integrity**: Cryptographic hashing ensures file authenticity
- **Cost Efficiency**: Only hash stored on-chain, reducing blockchain storage costs
- **Availability**: Files accessible as long as pinned on IPFS nodes

# Consensus Algorithm

A consensus algorithm is a protocol used by distributed systems and blockchains to ensure that all network participants (nodes) agree on a single, valid version of the system's data—despite the presence of failures or malicious actors. Its primary function is to maintain integrity, prevent double spending, and enable decentralized trust.

# Why Is Consensus Essential?

- Agreement: Ensures all nodes record the same transactions and data.
- Security: Protects against fraud, conflicting records, and attacks.
- Decentralization: Operates without relying on a central authority.
- Fault Tolerance: Handles failures, network delays, or compromised participants.

# Polygon’s Proof of Stake (PoS)

- Validators lock up (stake) MATIC tokens.
- Selection of validators is proportional to their stake.
- Block Approval: Validators propose and validate new blocks.
- Security: Misbehavior (fraudulent validation) can cause loss of staked funds.
- Efficiency: Lower energy consumption and faster throughput than PoW.

How Consensus Fits in Blockchain Applications

Data Flow Steps:

1. Transaction Submission: A user’s request (e.g., storing AI model data) is broadcast to the network.
2. Validation: Nodes (validators/miners) verify the transaction follows protocol rules.
3. Block Formation: A valid transaction is bundled into a block proposal.
4. Network Agreement: The consensus mechanism determines whether to accept the block.
5. Chain Update: Once agreed, the block becomes part of an immutable chain.
6. Finality: Updates are finalized; the state is updated across all nodes.

# How Consensus Algorithm Fits in Blockchain Application

**Data Flow Steps:**

1. Transaction Submission: A user's request (e.g., storing AI model data) is broadcast to the network.

2. Validation: Nodes (validators/miners) verify the transaction follows protocol rules.

3. Block Formation: A valid transaction is bundled into a block proposal.

4. Network Agreement: The consensus mechanism determines whether to accept the block.

5. Chain Update: Once agreed, the block becomes part of an immutable chain.

6. Finality: Updates are finalized; the state is updated across all nodes.

**Key Benefits**

- Prevents Double Spending: Ensures each transaction is unique and non-duplicatable.
- Enables Decentralization: Makes it possible for anyone to participate, removing single points of control.
- Scales Trust: Allows global participants to agree on history without prior relationships.

.

# References

[1] A. Sharma, R. K. Kaushal, and N. Kumar, “A comprehensive assessment of blockchain technology in the healthcare industry and techniques to store medical image,” in 2024 Third International Conference on Electrical, Electronics, Information and Communication Technologies (ICEEICT), 2024, pp. 1–6, doi: 10.1109/ICEEICT61591.2024.10718467.

[2] Y. Fu, X. Niu, L. Zhou, X. Cai, F. R. Yu, N. Cheng, and C. Li, “A hierarchical blockchain-enabled secure aggregation algorithm for federated learning in IoV,” IEEE Internet of Things Journal, vol. 12, no. 5, pp. 5876–5889, Mar. 2025, doi: 10.1109/JIOT.2024.3489032.

[3] D. Bouslimi, G. Coatrieux, M. Cozic, and C. Roux, “A joint encryption/watermarking system for verifying the reliability of medical images,” IEEE Transactions on Information Technology in Biomedicine, vol. 16, no. 5, pp. 891–899, Sep. 2012, doi: 10.1109/TITB.2012.2207730.

[4] P. Liu, Q. He, Y. Chen, S. Jiang, B. Zhao, and X. Wang, “A lightweight authentication and privacy-preserving aggregation for blockchain-enabled federated learning in VANETs,” IEEE Transactions on Consumer Electronics, vol. 71, no. 1, pp. 1274–1287, Feb. 2025,

[5] J. Li, J. Sun, F. Li, X. Yang, and Y. Wang, “BIFLC: A blockchain and IPFS-based multi-consensus federated learning framework,” in 2023 26th International Conference on Computer Supported Cooperative Work in Design (CSCWD), 2023, pp. 1–6,

# References

[6] J. Indumathi, A. Shankar, M. R. Ghalib, J. Gitanjali, Q. Hua, Z. Wen, and X. Qi, “Block chain-based Internet of Medical Things for uninterrupted, ubiquitous, user-friendly, unflappable, unblemished, unlimited health care services (BC IoMT U6 HCS),” IEEE Access, vol. 8, pp. 216856–216871, 2020, doi: 10.1109/ACCESS.2020.3040240.

[7] B. C. Priya, M. Dhanalakshmi, B. T. Kumar Reddy, A. Rohith and B. R. Reddy, "Block chain Based Inter-Organizational Secure File Sharing System," 2024 International Conference on Computational Intelligence for Security, Communication and Sustainable Development (CISCSD), Port Blair, India, 2024, pp. 230-235, doi: 10.1109/CISCSD63381.2024.00059.

[8] N. Nezhadsistani, N. S. Moayedian, and B. Stiller, “Blockchain-enabled federated learning in healthcare: Survey and state-of-the-art,” IEEE Access, vol. 13, pp. 119922–119941, 2025

[9] R. Kumar, A. A. Khan, J. Kumar, Zakria, N. A. Golilarz, S. Zhang, Y. Ting, C. Zheng, and W. Wang, “Blockchain-federated-learning and deep learning models for COVID-19 detection using CT imaging,” IEEE Sensors Journal, vol. 21, no. 14, pp. 16301–16314, Jul. 2021, doi: 10.1109/JSEN.2021.3076767.

# Thank You!

Photo by Pexels