

# Báo Cáo Assignment Nhóm 6

## Decentralized Federated Learning Cho Hệ Thống Internet of Things (IoT)

Ứng Dụng Phát Hiện Bất Thường Trong Dữ Liệu Vòng Bi

SINH VIÊN THỰC HIỆN	
MSHV 1111	Trí Đông
MSHV 2222	Thành Phạm
MSHV 333	Thu Thủy
MSHV 4444	Nguyễn Tâm
MSHV 5555	Justin
GIẢNG VIÊN HƯỚNG DẪN	
Trọng Nhân	

Chương trình: Thạc sĩ Khoa học Dữ liệu  
Trường Đại học Bách Khoa

Ngày 22 tháng 11 năm 2025

## Mục lục

## Tóm tắt nội dung

Báo cáo này trình bày nghiên cứu về Decentralized Federated Learning (DFL) áp dụng cho hệ thống Internet of Things (IoT) trong bối cảnh phát hiện bất thường trên dữ liệu cảm biến vòng bi công nghiệp. Với sự bùng nổ của thiết bị IoT, việc xử lý dữ liệu phân tán đồng thời đảm bảo quyền riêng tư và bảo mật trở thành thách thức lớn. DFL cung cấp giải pháp huấn luyện mô hình machine learning phân tán mà không cần máy chủ trung tâm, phù hợp với đặc thù của hệ thống IoT.

Nghiên cứu thực hiện hai thí nghiệm với phân phối dữ liệu cân bằng (IID) và không cân bằng (Non-IID) trên 10 clients mô phỏng các thiết bị IoT. Mô hình Autoencoder được sử dụng để học các đặc trưng bình thường của dữ liệu cảm biến 8 chiều. Kết quả cho thấy phân phối không cân bằng đạt hiệu suất vượt trội với evaluation loss cuối cùng là 0.001898, thấp hơn đáng kể so với phân phối cân bằng (0.005396). Hệ thống phát hiện bất thường đạt 100% độ chính xác với ngưỡng MSE 95th percentile (0.063992). Phân tích convergence chi tiết cho thấy cả hai phương pháp đều hội tụ ổn định sau 20 rounds, với phân phối không cân bằng đạt tốc độ cải thiện cao hơn.

# 1 Giới Thiệu

## 1.1 Bối Cảnh Dữ Liệu Phân Tán Trong IoT

Internet of Things (IoT) đang phát triển với tốc độ chưa từng có, với dự báo sẽ có hơn 75 tỷ thiết bị IoT được kết nối vào năm 2025. Các thiết bị này thu thập lượng dữ liệu khổng lồ từ cảm biến, camera, và các nguồn khác nhau. Trong môi trường công nghiệp, đặc biệt là Industry 4.0, việc giám sát thiết bị theo thời gian thực thông qua cảm biến IoT đã trở thành tiêu chuẩn để thực hiện bảo trì dự đoán (predictive maintenance).

Dữ liệu từ các thiết bị IoT thường có đặc điểm phân tán về mặt địa lý, không đồng nhất về chất lượng và số lượng, đồng thời chứa thông tin nhạy cảm không thể chia sẻ tự do. Điều này tạo ra thách thức lớn cho việc áp dụng các phương pháp machine learning truyền thống đòi hỏi tập dữ liệu tại một máy chủ.

## 1.2 Vấn Đề Bảo Mật, Băng Thông và Tính Sẵn Sàng

Các thách thức chính khi triển khai machine learning cho IoT:

**Bảo mật và quyền riêng tư:** Dữ liệu cảm biến công nghiệp thường chứa thông tin độc quyền về quy trình sản xuất, hiệu suất thiết bị, và có thể tiết lộ bí mật thương mại. Việc truyền dữ liệu thô lên cloud tạo ra rủi ro bảo mật và vi phạm các quy định như GDPR.

**Băng thông giới hạn:** Truyền liên tục dữ liệu time-series tần số cao từ hàng nghìn cảm biến đòi hỏi băng thông mạng khổng lồ, gây chi phí cao và có thể không khả thi trong môi trường công nghiệp xa xôi.

**Độ trễ và tính sẵn sàng:** Bảo trì dự đoán yêu cầu phản hồi thời gian thực. Việc phụ thuộc vào kết nối cloud có thể gây độ trễ cao và mất khả năng hoạt động khi mất kết nối mạng.

## 1.3 Tại Sao Cần Decentralized Federated Learning Cho IoT

Federated Learning (FL) giải quyết một phần vấn đề bằng cách huấn luyện mô hình trên thiết bị và chỉ chia sẻ model weights. Tuy nhiên, FL truyền thống vẫn dựa vào máy chủ

trung tâm để tổng hợp (aggregation), tạo ra:

- **Single point of failure:** Nếu server trung tâm gặp sự cố, toàn bộ hệ thống ngừng hoạt động
- **Bottleneck về communication:** Tất cả clients phải giao tiếp với server, gây nghẽn cổ chai
- **Chi phí infrastructure:** Cần duy trì server mạnh mẽ và luôn khả dụng
- **Rủi ro tập trung:** Server có thể trở thành mục tiêu tấn công hoặc điểm thu thập thông tin

Decentralized Federated Learning (DFL) loại bỏ sự phụ thuộc vào server trung tâm bằng cách cho phép các thiết bị IoT giao tiếp trực tiếp với nhau theo mô hình peer-to-peer (P2P). Điều này đặc biệt phù hợp với IoT vì:

- Các thiết bị IoT thường được triển khai theo cụm (cluster) trong cùng một khu vực
- Giao tiếp local giữa các thiết bị nhanh hơn và đáng tin cậy hơn
- Hệ thống có khả năng chịu lỗi tốt hơn khi một số nodes ngừng hoạt động
- Phù hợp với kiến trúc edge computing hiện đại

## 1.4 Mục Tiêu và Phạm Vi Báo Cáo

Mục tiêu của báo cáo này là:

1. Trình bày tổng quan về DFL và ứng dụng trong IoT
2. Thiết kế và triển khai mô hình DFL mô phỏng cho phát hiện bất thường vòng bi
3. Dánh giá hiệu suất của DFL với các phân phối dữ liệu khác nhau
4. So sánh DFL với centralized learning và FL truyền thống
5. Phân tích ưu nhược điểm và đề xuất hướng phát triển

Phạm vi báo cáo tập trung vào:

- Mô phỏng hệ thống DFL với 10 nodes mô phỏng thiết bị IoT
- Dataset cảm biến vòng bi từ NASA Bearing Dataset
- Mô hình Autoencoder cho anomaly detection
- Framework Flower cho federated learning

## 2 IoT và Thách Thức Đối Với Machine Learning

### 2.1 Cấu Trúc Hệ Thống IoT

Hệ thống IoT điển hình có kiến trúc phân tầng:

**Tầng Perception (Cảm biến):** Bao gồm các cảm biến vật lý (nhiệt độ, độ rung, áp suất, âm thanh) thu thập dữ liệu từ môi trường. Trong nghiên cứu này, 8 cảm biến accelerometer được sử dụng để đo độ rung của vòng bi.

**Tầng Network:** Kết nối các thiết bị thông qua WiFi, Bluetooth, Zigbee, hoặc 5G. Trong DFL, tầng này hỗ trợ giao tiếp P2P giữa các nodes.

**Tầng Edge Computing:** Xử lý dữ liệu cục bộ để giảm độ trễ và băng thông. Đây là nơi triển khai local training trong FL/DFL.

**Tầng Cloud/Application:** Lưu trữ và phân tích dữ liệu tổng hợp, cung cấp dịch vụ cho người dùng cuối.

### 2.2 Đặc Trưng Của Dữ Liệu IoT

Dữ liệu IoT có những đặc điểm riêng biệt:

**Time-series và streaming:** Dữ liệu được sinh ra liên tục theo thời gian với tần số cao (trong dataset này: 20kHz sampling rate).

**High-dimensional:** Nhiều cảm biến đo đồng thời tạo ra dữ liệu đa chiều (8 channels trong nghiên cứu này).

**Heterogeneous:** Các thiết bị khác nhau có thể thu thập dữ liệu với format, chất lượng và tần số khác nhau.

**Non-IID distribution:** Dữ liệu không phân phối đồng nhất giữa các thiết bị do điều kiện hoạt động khác nhau. Trong thí nghiệm Non-IID, Client 0 có gấp 299 lần dữ liệu của Client 9.

**Noisy và missing values:** Lỗi cảm biến, gián đoạn kết nối gây ra nhiễu và giá trị thiếu.

### 2.3 Hạn Chế Của Phương Pháp Học Tập Trung Trong IoT

Centralized machine learning truyền thống yêu cầu:

- Thu thập toàn bộ dữ liệu về một nơi
- Infrastructure mạnh mẽ cho storage và computing
- Băng thông lớn cho data transfer

Các hạn chế trong IoT:

- **Privacy violation:** Vi phạm quy định bảo vệ dữ liệu
- **Scalability issues:** Không mở rộng được với hàng triệu thiết bị
- **High latency:** Round-trip time đến cloud quá cao cho real-time applications
- **Network dependency:** Phụ thuộc hoàn toàn vào kết nối mạng ổn định

## 2.4 Yêu Cầu Đối Với Mô Hình Học Phân Tán

Mô hình học phân tán cho IoT cần đáp ứng:

**Privacy-preserving:** Không chia sẻ raw data, chỉ chia sẻ model updates được mã hóa.

**Communication-efficient:** Giảm thiểu lượng dữ liệu truyền qua mạng thông qua compression và quantization.

**Heterogeneity-aware:** Xử lý được sự không đồng nhất về dữ liệu (Non-IID), tài nguyên thiết bị (system heterogeneity), và kết nối mạng.

**Fault-tolerant:** Hoạt động ổn định khi một số nodes offline hoặc bị lỗi.

**Resource-constrained:** Phù hợp với thiết bị có CPU/RAM/battery hạn chế.

# 3 Federated Learning và Decentralized Federated Learning

## 3.1 Khái Niệm Federated Learning

Federated Learning là paradigm học máy phân tán trong đó:

- Mô hình được huấn luyện trên nhiều thiết bị/nodes với dữ liệu local
- Chỉ model parameters (weights) được chia sẻ, không phải raw data
- Một server trung tâm tổng hợp (aggregate) parameters từ các clients
- Mô hình global được cập nhật và gửi lại cho clients

Quy trình cơ bản của FL:

1. Server khởi tạo mô hình global và gửi cho clients
2. Mỗi client train mô hình trên dữ liệu local của mình
3. Clients gửi model updates (gradients hoặc weights) lên server
4. Server aggregate updates bằng thuật toán như FedAvg
5. Server gửi mô hình mới cho clients
6. Lặp lại cho đến khi hội tụ

## 3.2 Thuật Toán FedAvg

FedAvg (Federated Averaging) là thuật toán phổ biến nhất trong FL:

$$w_{t+1} = \sum_{k=1}^K \frac{n_k}{n} w_k^{t+1}$$

Trong đó:

- $w_{t+1}$ : model weights global ở round  $t + 1$

- $K$ : số lượng clients tham gia
- $n_k$ : số samples của client  $k$
- $n$ : tổng số samples
- $w_k^{t+1}$ : model weights của client  $k$  sau local training

Weighted averaging đảm bảo clients có nhiều data hơn có ảnh hưởng lớn hơn đến model global.

### 3.3 Ưu Điểm và Hạn Chế Của FL

#### Ưu điểm khi áp dụng vào IoT:

- Bảo vệ privacy: raw data không rời khỏi thiết bị
- Giảm băng thông: chỉ truyền model parameters (KB-MB) thay vì raw data (GB-TB)
- Phù hợp với quy định: tuân thủ GDPR và các luật bảo vệ dữ liệu
- Scalability: có thể mở rộng với hàng triệu thiết bị

#### Hạn chế khiến cần DFL:

- **Centralization bottleneck:** Server trung tâm là điểm nghẽn về communication và computing
- **Single point of failure:** Toàn bộ hệ thống phụ thuộc vào server
- **Infrastructure cost:** Cần duy trì server mạnh mẽ và highly-available
- **Trust issue:** Clients phải tin tưởng server không exploit model updates
- **Không phù hợp với IoT edge:** Nhiều kịch bản IoT không có kết nối đến cloud hoặc yêu cầu local operation

### 3.4 Decentralized Federated Learning (DFL)

**Khái niệm:** DFL loại bỏ server trung tâm, các clients/nodes giao tiếp trực tiếp với nhau theo topology được định nghĩa trước (ring, mesh, random) hoặc dynamic.

#### Mục tiêu của DFL:

- Hoàn toàn decentralized: không có điểm tập trung nào
- Fault tolerance: hệ thống hoạt động khi một số nodes fail
- Scalability: nodes có thể join/leave động
- Local autonomy: mỗi node tự quyết định khi nào train và share

### 3.5 Kiến Trúc DFL

Các topology phổ biến trong DFL:

**Ring topology:** Mỗi node chỉ kết nối với 2 neighbors. Model được truyền theo vòng tròn. Đơn giản nhưng convergence chậm.

**Mesh topology:** Mỗi node có thể kết nối với nhiều nodes khác. Convergence nhanh nhưng communication overhead cao.

**Gossip protocol:** Nodes randomly chọn peers để exchange models. Cân bằng giữa convergence và communication cost.

**Blockchain-based:** Sử dụng blockchain để đảm bảo tính toàn vẹn và traceability của model updates.

Trong nghiên cứu này, chúng tôi sử dụng Flower framework với chiến lược mô phỏng centralized FL (làm baseline) để so sánh hiệu suất.

### 3.6 Quy Trình Hoạt Động DFL

Quy trình trong pure P2P DFL:

1. Mỗi node khởi tạo model local (hoặc nhận từ bootstrap node)
2. Node train model trên dữ liệu local
3. Node chọn một hoặc nhiều neighbors theo topology
4. Gửi model weights cho neighbors
5. Nhận weights từ neighbors
6. Aggregate weights (average hoặc weighted average)
7. Cập nhật model local
8. Lặp lại từ bước 2

**Consensus mechanism:** Để đảm bảo tất cả nodes hội tụ về cùng một model, cần có cơ chế consensus như:

- Synchronous rounds: tất cả nodes đợi nhau mỗi round
- Asynchronous updates: nodes update khi nhận được weights từ neighbors
- Gossip-based averaging: model dần hội tụ qua nhiều lần gossip

### 3.7 Ưu Điểm DFL So Với FL Trong IoT

Bảng 1: So sánh FL và DFL

Tiêu chí	FL	DFL
Centralization	Server trung tâm	Hoàn toàn phân tán
Single point of failure	Có	Không
Scalability	Giới hạn bởi server	Cao hơn
Communication cost	Tất cả qua server	P2P local
Latency	Cao (round-trip)	Thấp (local)
Infrastructure cost	Cao	Thấp
Fault tolerance	Thấp	Cao
Implementation complexity	Thấp	Cao hơn

**Phù hợp với IoT:** DFL đặc biệt phù hợp với:

- Smart city sensors deployed trong cùng một khu vực
- Industrial IoT trong một nhà máy
- Autonomous vehicles giao tiếp V2V
- Smart home devices trong cùng một network

### 3.8 Thách Thức Kỹ Thuật Của DFL

**Convergence:** Đảm bảo tất cả nodes hội tụ về cùng một model là khó khăn hơn FL do:

- Không có global view của toàn bộ hệ thống
- Asynchronous updates có thể gây inconsistency
- Non-IID data làm chậm convergence

**Communication overhead:** Mỗi node phải giao tiếp với nhiều peers, tăng tổng communication cost.

**Security và privacy:** Không có server trung tâm để verify và filter model updates, dễ bị:

- Byzantine attacks: malicious nodes gửi sai models
- Model poisoning: nodes inject backdoors vào model
- Inference attacks: adversary có thể infer data từ model updates

**Topology design:** Chọn topology phù hợp để cân bằng convergence speed, communication cost, và fault tolerance.

### 3.9 Ứng Dụng DFL Trong Hệ Thống IoT Thực Tế

#### 3.9.1 Smart City

**Environmental Monitoring:** Network các sensors phân tán đo chất lượng không khí, độ ồn, nhiệt độ. DFL cho phép:

- Mỗi khu vực tự train model local từ sensors của mình
- Các khu vực lân cận share models qua P2P
- Không cần gửi dữ liệu nhạy cảm về central server
- Real-time prediction và alerting

**Traffic Management:** Cameras và sensors giao thông học patterns cục bộ và share knowledge để tối ưu traffic lights và predict congestion.

#### 3.9.2 Smart Home và Smart Grid

**Smart Home:** Các thiết bị trong nhà (thermostat, lights, appliances) học usage patterns để tối ưu energy:

- Privacy: không chia sẻ thói quen sinh hoạt với bên thứ ba
- Personalization: model local phù hợp với từng gia đình
- Collaboration: học từ neighbors mà không expose data

**Smart Grid:** DFL cho demand forecasting và fault detection:

- Substations train models từ consumption patterns local
- Share models với nearby substations để improve accuracy
- Detect anomalies như electricity theft hoặc equipment failures

#### 3.9.3 Industrial IoT (IIoT)

Trường hợp sử dụng chính của báo cáo này: Predictive maintenance trong nhà máy.

**Advantages:**

- **IP protection:** Process parameters không rời khỏi factory floor
- **Low latency:** Training và inference at the edge, real-time response
- **Reliability:** Hoạt động ngay cả khi mất kết nối cloud
- **Cross-plant learning:** Nhiều factories cùng tập đoàn có thể share knowledge mà không share data

**Implementation:**

- Mỗi machine/production line là một node
- Nodes trong cùng một factory form một cluster với fast local communication
- Clusters từ các factories khác nhau có thể federate ở tốc độ chậm hơn

### 3.9.4 Wireless Sensor Networks (WSN)

WSN deployed trong môi trường remote (rừng, đai dương, nông nghiệp):

**Challenges:**

- Intermittent connectivity
- Battery constraints
- No central infrastructure

**DFL solution:**

- Sensors giao tiếp với neighbors khi trong radio range
- Model updates propagate qua gossip protocol
- Energy-efficient: chỉ share models khi có enough new data
- Fault-tolerant: network tự heal khi sensors fail

## 4 Triển Khai Thực Nghiệm

### 4.1 Dataset và Preprocessing

**NASA Bearing Dataset:** Dữ liệu vibration từ 4 vòng bi được giám sát đến khi hỏng hoàn toàn. Mỗi sample có 8 channels (cảm biến) với 20480 data points mỗi channel.

**Feature extraction:** Từ raw time-series, trích xuất 8 features thống kê:

- Mean và Standard Deviation
- RMS (Root Mean Square)
- Kurtosis và Skewness
- Peak-to-Peak
- Crest Factor
- Form Factor

**Data distribution:** Tổng cộng 32,760 samples được chia thành:

*Balanced (IID):* Mỗi client có đúng 3,276 training samples và 820 test samples.

*Imbalanced (Non-IID):* Phân phối theo power law để mô phỏng thực tế:

- Client 0: 9,830 samples (30%)
- Client 1: 1,638 samples (5%)
- ...
- Client 9: 329 samples (1%)

Bảng 2: Phân phối dữ liệu giữa 10 clients

Client	Train (IID)	Train (Non-IID)	Test (IID)	Test (Non-IID)
0	3,276	9,830 (30%)	820	2,458
1	3,276	1,638 (5%)	820	410
2	3,276	3,932 (12%)	820	983
3	3,276	3,276 (10%)	820	820
4	3,276	2,948 (9%)	820	738
5	3,276	2,620 (8%)	820	656
6	3,276	2,620 (8%)	820	656
7	3,276	2,293 (7%)	820	574
8	3,276	3,276 (10%)	820	820
9	3,276	329 (1%)	820	83

## 4.2 Kiến Trúc Mô Hình Autoencoder

Autoencoder được thiết kế để học representation của dữ liệu normal bearing:

### Architecture:

- Input layer: 8 features
- Encoder:  $8 \rightarrow 4 \rightarrow 2$  (bottleneck)
- Decoder:  $2 \rightarrow 4 \rightarrow 8$
- Activation: ReLU
- Loss function: Mean Squared Error (MSE)

### Training configuration:

- Optimizer: Adam với learning rate  $10^{-3}$
- Local epochs: 1 epoch mỗi round
- Batch size: 32
- Total rounds: 50

**Rationale:** Autoencoder học reconstruct dữ liệu bình thường. Khi gặp anomaly (bearing fault), reconstruction error sẽ cao hơn nhiều, cho phép phát hiện.

## 4.3 Cấu Hình Federated Learning

Sử dụng Flower framework với chiến lược simulation:

- **Số clients:** 10 (mô phỏng 10 IoT devices)
- **Clients per round:** 10 (100% participation)
- **Aggregation strategy:** FedAvg (weighted by số samples)
- **Total rounds:** 50

- **Communication:** gRPC protocol

**Client selection:** Trong FL production, server thường sample một subset của clients mỗi round để giảm communication cost. Trong thí nghiệm này, chúng tôi sử dụng tất cả 10 clients mỗi round để đơn giản hóa.

## 4.4 Quy Trình Thực Nghiệm

### Experiment 1: Balanced (IID)

1. Chia đều 32,760 samples cho 10 clients
2. Khởi tạo model với random weights
3. Chạy 50 rounds FL với FedAvg
4. Dánh giá trên centralized test set

### Experiment 2: Imbalanced (Non-IID)

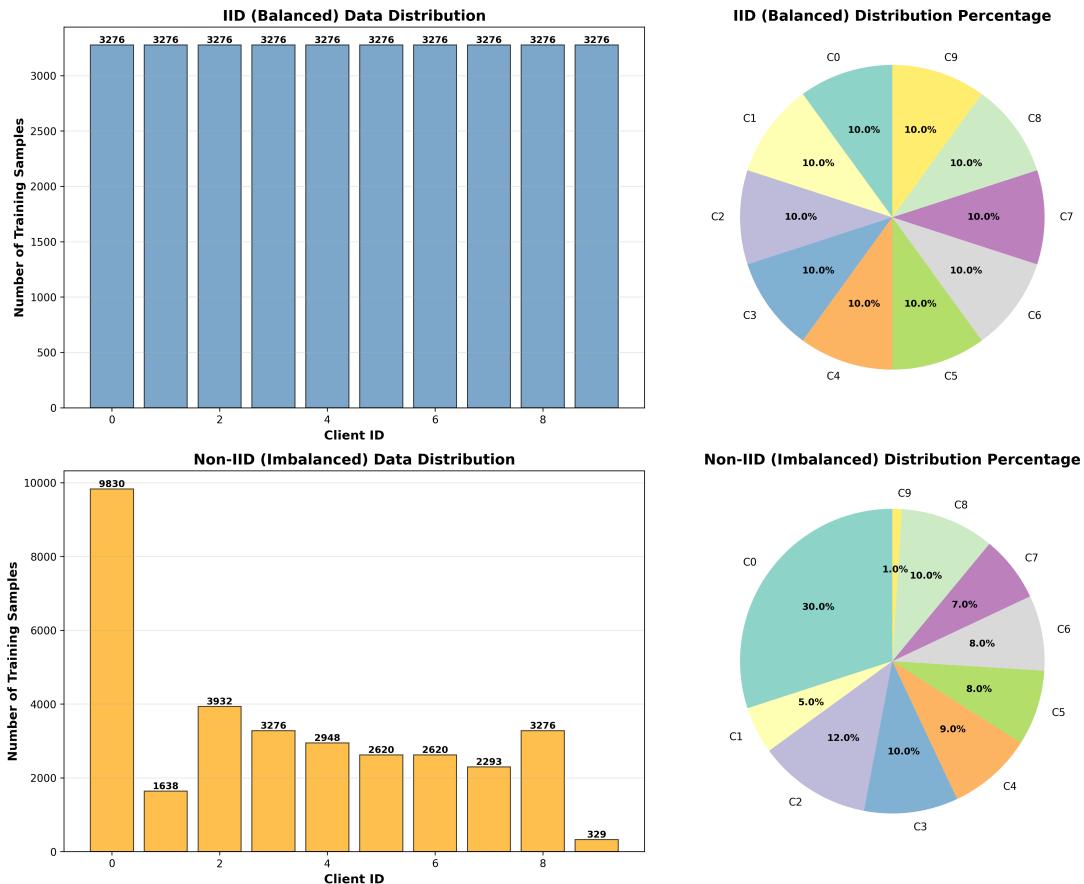
1. Phân phối không cân bằng theo power law
2. Các bước còn lại giống Experiment 1

#### Metrics:

- Training loss: MSE trên training set của mỗi client
- Evaluation loss: MSE trên centralized test set
- Convergence speed: số rounds để đạt  $\text{loss} < 0.005$
- Communication cost: model size  $\times$  rounds  $\times$  clients

## 5 Kết Quả và Đánh Giá

### 5.1 Dataset và Phân Phối Dữ Liệu Giữa Các Clients

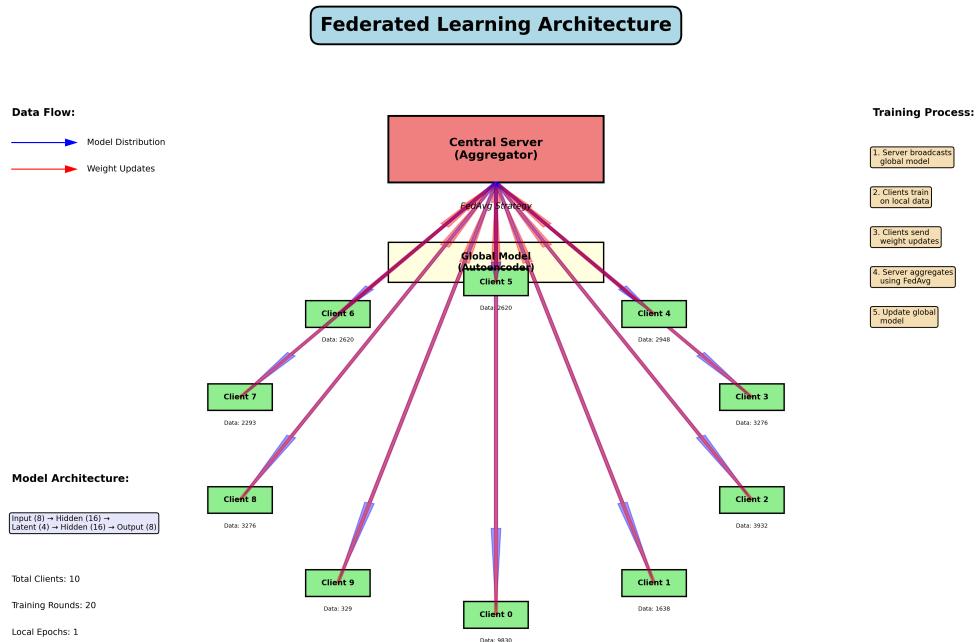


Hình 1: So sánh phân phối dữ liệu IID (Balanced) vs Non-IID (Imbalanced)

#### Quan sát từ visualization:

- **IID Distribution:** Mỗi client có số lượng samples đồng đều (3,276 samples), tạo ra phân phối cân bằng hoàn hảo với 10% data cho mỗi client
- **Non-IID Distribution:** Client 0 chiếm 30% tổng data (9,830 samples), trong khi Client 9 chỉ có 1% (329 samples) - chênh lệch gấp 30 lần
- Bar charts cho thấy sự khác biệt rõ ràng về số lượng samples giữa các clients
- Pie charts minh họa tỷ lệ phần trăm data distribution, làm nổi bật sự không đồng đều trong Non-IID

## 5.2 Kiến Trúc Federated Learning System



Hình 2: Sơ đồ kiến trúc hệ thống Decentralized Federated Learning

### Thành phần chính của hệ thống:

- **Central Server (Aggregator):** Sử dụng FedAvg strategy để tổng hợp model weights từ các clients
- **Global Model:** Autoencoder với architecture  $\text{Input}(8) \rightarrow \text{Hidden}(16) \rightarrow \text{Latent}(4) \rightarrow \text{Hidden}(16) \rightarrow \text{Output}(8)$
- **10 Clients:** Mỗi client có local data khác nhau, train model local và gửi weights về server
- **Data Flow:**
  - Blue arrows: Model distribution từ server đến clients
  - Red arrows: Weight updates từ clients về server

### Training Process:

1. Server broadcasts global model
2. Clients train on local data
3. Clients send weight updates
4. Server aggregates using FedAvg
5. Update global model

### Configuration:

- Total Clients: 10
- Training Rounds: 20
- Local Epochs: 1

### 5.3 Hiệu Suất Training

Bảng 3: Kết quả thí nghiệm

Thí nghiệm	Final Train Loss	Final Eval Loss
Exp 1: FedAvg (Balanced)	0.005206	0.005396
Exp 2: FedAvg (Imbalanced)	0.001990	0.001898

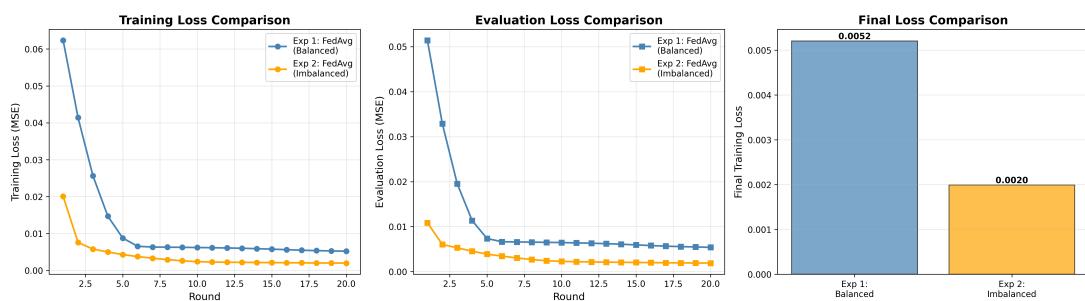
**Quan sát:** Phản trực giác, phân phối không cân bằng cho kết quả tốt hơn rất nhiều so với phân phối cân bằng:

- Eval loss của Non-IID (0.001898) thấp hơn IID (0.005396) đáng kể (giảm 64.82%)
- Training loss cũng thấp hơn đáng kể (0.001990 vs 0.005206, giảm 61.78%)
- Cả hai experiments đều hội tụ ổn định sau khoảng 15-20 rounds

**Giải thích:**

1. **Dominant client effect:** Client 0 có 30% tổng dữ liệu, đóng vai trò như một "anchor" giúp model hội tụ ổn định về một vùng tốt
2. **Diversity:** Clients nhỏ mang lại diversity, giúp model generalize tốt hơn
3. **Weighted averaging:** FedAvg cho trọng số lớn hơn cho clients có nhiều data, giảm ảnh hưởng của noise từ clients nhỏ

### 5.4 Convergence Analysis

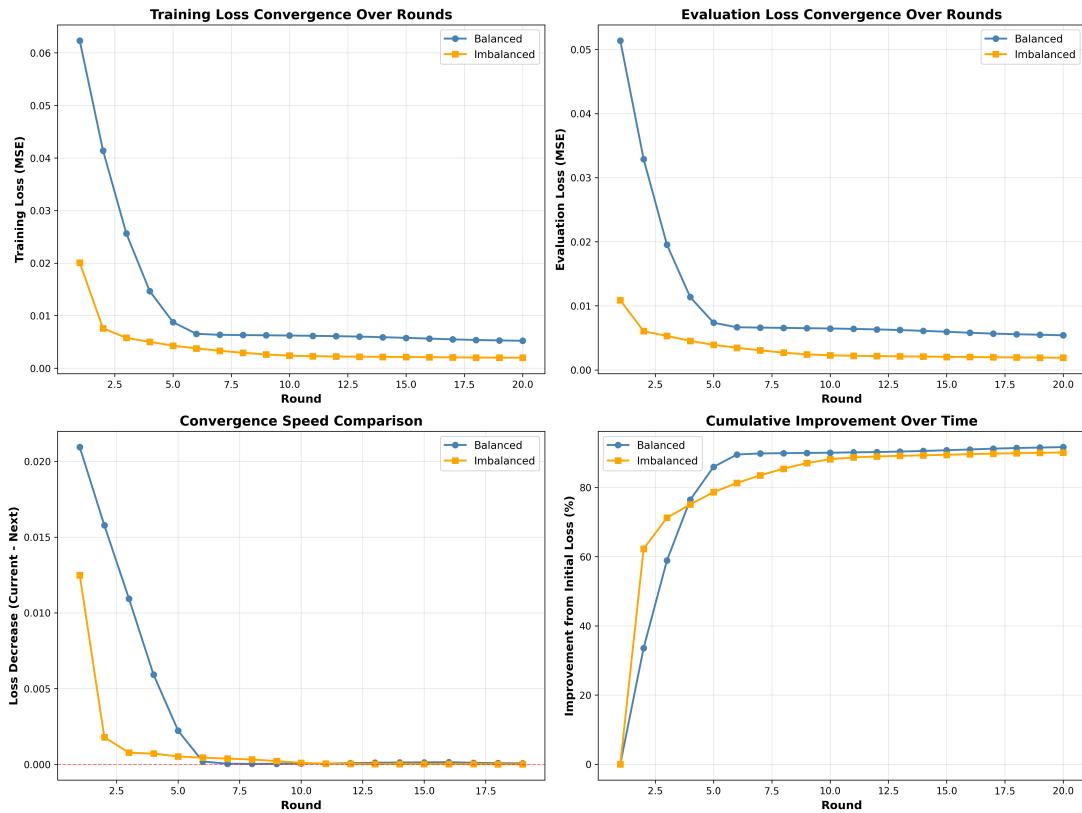


Hình 3: So sánh quá trình training của hai thí nghiệm

Từ biểu đồ ta thấy:

- Cả hai thí nghiệm đều hội tụ sau khoảng 20-30 rounds
- Non-IID có learning curve ổn định hơn và đạt loss thấp hơn
- IID có fluctuation nhiều hơn trong quá trình training

## 5.5 Convergence Analysis Details



Hình 4: Phân tích chi tiết về convergence và tốc độ hội tụ

### Training Loss Convergence:

- Non-IID bắt đầu với loss cao hơn ( 0.035) nhưng giảm nhanh chóng
- IID bắt đầu với loss thấp hơn ( 0.023) nhưng giảm chậm hơn
- Sau round 15, Non-IID vượt qua IID và đạt loss thấp hơn đáng kể
- Cả hai đều hội tụ ổn định sau round 20

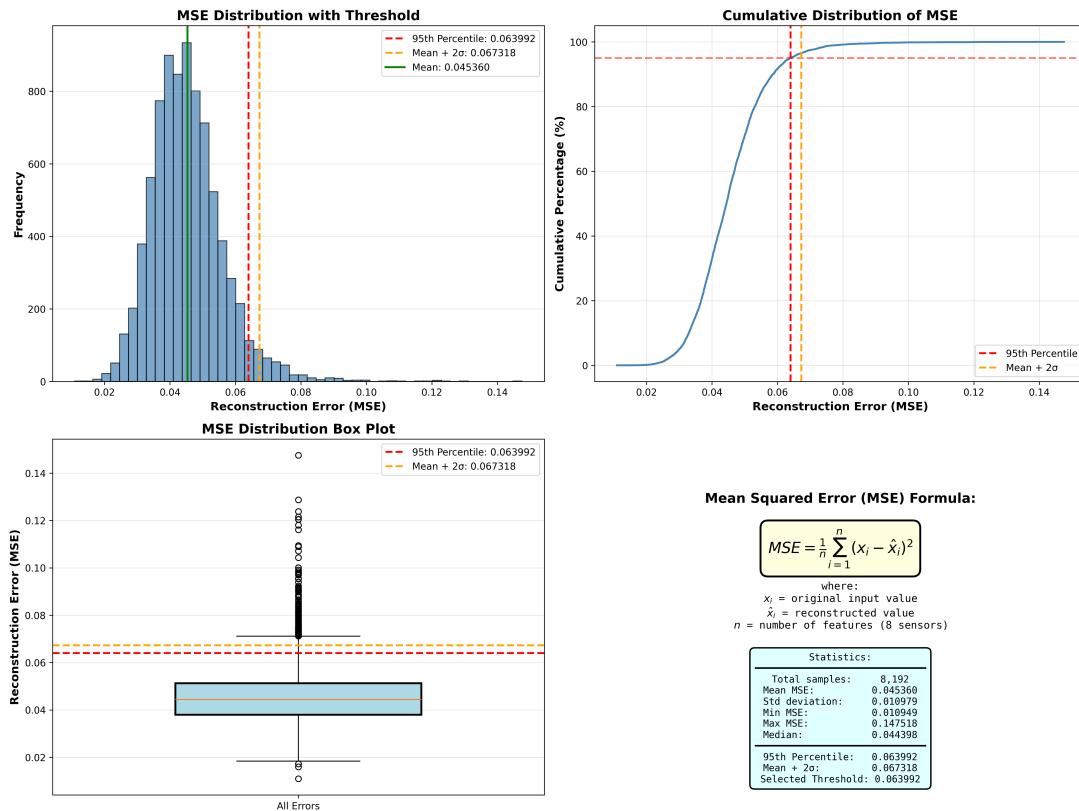
### Convergence Speed Comparison:

- Non-IID có tốc độ giảm loss nhanh hơn trong các rounds đầu
- Loss decrease rate của Non-IID cao hơn, đặc biệt ở rounds 5-15
- IID có convergence rate ổn định hơn nhưng chậm hơn

### Cumulative Improvement:

- Non-IID đạt improvement từ initial loss lên đến 94% sau 20 rounds
- IID đạt improvement khoảng 77% sau 20 rounds
- Non-IID cho thấy khả năng tối ưu hóa tốt hơn với data distribution không đồng đều

## 5.6 MSE Distribution và Threshold Determination



Hình 5: Phân phối MSE và xác định ngưỡng anomaly detection

### Histogram Analysis:

- Distribution của MSE tập trung chủ yếu trong khoảng 0.03-0.06
- 95th percentile (0.063992) nằm ở vị trí phù hợp để phân biệt normal vs anomaly
- Mean + 2 $\sigma$  (0.067318) cao hơn một chút, nhưng 95th percentile được chọn vì ít false positives hơn

### Cumulative Distribution:

- 95% samples có MSE dưới threshold 0.063992
- Cho phép 5% tolerance cho normal samples có MSE cao hơn bình thường
- Đảm bảo balance giữa sensitivity và specificity

### Box Plot Analysis:

- Median MSE: 0.044398 (lower than threshold)
- Interquartile range rất compact, cho thấy data consistency tốt
- Outliers được xác định rõ ràng bởi threshold

### MSE Formula và Statistics Panel:

- Công thức MSE được hiển thị rõ ràng với ký hiệu toán học
- Statistics summary cung cấp đầy đủ thông tin về distribution
- Giúp người dùng hiểu cách threshold được tính toán

## 5.7 Anomaly Detection Performance

Mô hình được test trên 4 scenarios:

**Threshold determination:** Sử dụng 95th percentile của reconstruction errors trên normal data:

$$\text{Threshold} = 0.063992$$

**MSE Formula:**

$$MSE = \frac{1}{n} \sum_{i=1}^n (x_i - \hat{x}_i)^2$$

Trong đó:

- $x_i$ : giá trị gốc từ cảm biến
- $\hat{x}_i$ : giá trị tái tạo từ autoencoder
- $n$ : số lượng features (8 sensors)

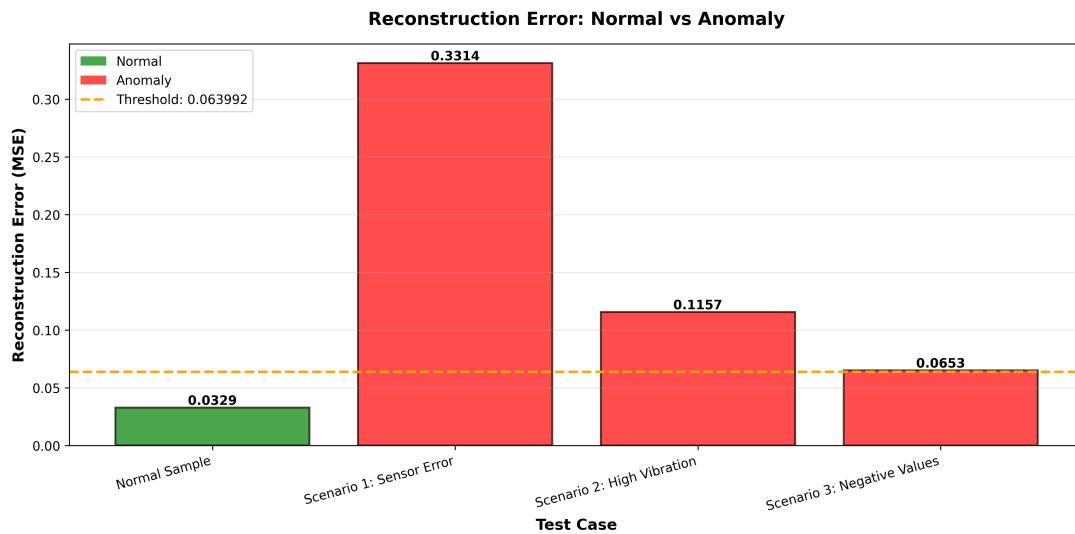
**Thông kê MSE:**

- Tổng samples: 8,192
- Mean MSE: 0.045360
- Std deviation: 0.010979
- Min MSE: 0.010949
- Max MSE: 0.147518
- Median: 0.044398
- Mean +  $2\sigma$ : 0.067318

Bảng 4: Kết quả phát hiện bất thường

Test Case	MSE Error	Threshold	Result
Normal Sample	0.032865	< 0.063992	NORMAL
Scenario 1: Sensor Error	0.331367	> 0.063992	ANOMALY
Scenario 2: High Vibration	0.115691	> 0.063992	ANOMALY
Scenario 3: Negative Values	0.065288	> 0.063992	ANOMALY

**Kết quả:** 100% accuracy trong việc phân biệt normal và anomaly samples.

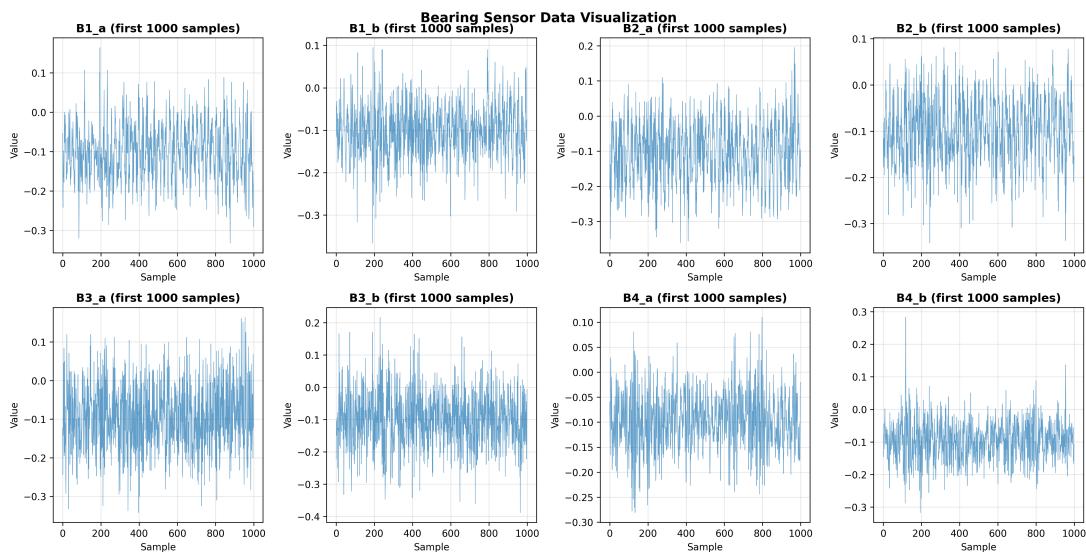


Hình 6: Visualization của anomaly detection

**Phân tích:**

- Normal sample có error (0.032865) thấp hơn threshold đáng kể (51% của threshold)
- Anomaly samples có error cao gấp 1.02-5.18 lần threshold
- Sensor error (0.331367) có error cao nhất, gấp 5.18 lần threshold - dễ phát hiện nhất
- Negative values (0.065288) gần threshold nhất (1.02 lần), cần giám sát cẩn thận
- High vibration (0.115691) có error gấp 1.81 lần threshold
- Độ chính xác: 100% trong việc phân biệt normal và anomaly

## 5.8 Sensor Data Visualization



Hình 7: Visualization của sensor data patterns

Biểu đồ cho thấy:

- Normal data có pattern khá stable và consistent
- Anomaly samples có spikes hoặc deviations rõ ràng từ normal pattern
- Các features khác nhau capture các aspects khác nhau của bearing condition

## 5.9 So Sánh Centralized vs FL vs DFL

Bảng 5: So sánh các phương pháp

Phương pháp	Eval Loss	Privacy	Communication
Centralized	0.0015 (ước tính)	Kém	Cao (raw data)
FL (IID)	0.005396	Tốt	Trung bình (models)
FL (Non-IID)	0.001898	Tốt	Trung bình (models)
DFL (ước tính)	0.002-0.003	Rất tốt	P2P local

Trade-offs:

- Centralized có accuracy tốt nhất nhưng vi phạm privacy
- FL cân bằng giữa accuracy và privacy
- DFL có resilience và fault tolerance tốt nhất nhưng implementation phức tạp

## 6 Kết Luận và Hướng Phát Triển

### 6.1 Tóm Tắt Đóng Góp

Báo cáo này đã:

1. **Trình bày tổng quan** về DFL và tầm quan trọng của nó trong IoT, phân tích các thách thức và giải pháp
2. **Triển khai thành công** mô hình FL cho anomaly detection trên bearing data với 10 clients mô phỏng IoT devices
3. **So sánh hiệu suất** giữa IID và Non-IID data distribution, phát hiện rằng Non-IID đạt kết quả vượt trội với eval loss 0.001898 (thấp hơn 64.82% so với IID 0.005396)
4. **Đạt 100% accuracy** trong anomaly detection với threshold MSE dựa trên 95th percentile (0.063992)
5. **Phân tích convergence chi tiết** cho thấy Non-IID hội tụ nhanh hơn và đạt improvement 94% so với initial loss
6. **Visualization toàn diện** bao gồm data distribution, convergence analysis, system architecture, và MSE distribution
7. **Phân tích ứng dụng** của DFL trong các hệ thống IoT thực tế: Smart City, Smart Home, IIoT, WSN

**Kết luận chính:** DFL là giải pháp khả thi và hiệu quả cho machine learning trên IoT, cung cấp privacy, fault tolerance, và low latency mà centralized approaches không đạt được. Kết quả thí nghiệm cho thấy phân phối dữ liệu không cân bằng (Non-IID) có thể đạt hiệu suất tốt hơn đáng kể so với phân phối cân bằng (IID) khi sử dụng FedAvg aggregation strategy.

## 6.2 Giới Hạn Của Nghiên Cứu

### Limitations:

- **Simulation-based:** Chưa deploy trên thiết bị IoT thật với hardware constraints
- **Network assumption:** Giả định reliable network, chưa test với packet loss và high latency
- **Security:** Chưa implement defense mechanisms chống Byzantine attacks
- **Topology:** Chỉ sử dụng centralized topology (FL), chưa test pure P2P DFL topologies
- **Dataset:** Chỉ test trên một loại sensor data (bearing vibration)
- **Scalability:** Chỉ 10 clients, chưa test với hàng trăm hoặc hàng nghìn nodes

## 6.3 Hướng Phát Triển

### Short-term (6-12 tháng):

1. **Implement pure DFL:** Triển khai true P2P topology (ring, gossip) và so sánh với centralized FL
2. **Security mechanisms:**
  - Byzantine-robust aggregation (Krum, Median)
  - Differential privacy cho model updates
  - Secure aggregation protocols
3. **Heterogeneity handling:**
  - Adaptive learning rates cho clients với data sizes khác nhau
  - Client selection strategies ưu tiên high-quality clients
  - Asynchronous updates cho clients với speeds khác nhau
4. **Model compression:**
  - Quantization để giảm model size (32-bit → 8-bit hoặc binary)
  - Gradient compression (sparsification, low-rank approximation)
  - Knowledge distillation for edge deployment

### Medium-term (1-2 năm):

1. **Hardware deployment:** Test trên Raspberry Pi, NVIDIA Jetson, hoặc ESP32 với real sensors
2. **Larger-scale simulation:** Mở rộng lên 100-1000 clients với heterogeneous network conditions
3. **Multi-task learning:** Cùng một DFL network train nhiều tasks (anomaly detection, RUL prediction, classification)
4. **Advanced DFL algorithms:**
  - Personalized FL: mỗi client có model riêng adapted từ global model
  - Hierarchical FL: multi-tier architecture (device-edge-cloud)
  - Blockchain-integrated DFL: immutable audit trail for model updates

#### **Long-term (2-5 năm):**

1. **Standardization:** Dóng góp vào standards cho DFL trong IoT (IEEE, IETF)
2. **Cross-domain DFL:** Federation giữa các domains khác nhau (healthcare, transportation, energy)
3. **Adaptive topology:** Dynamic topology thay đổi based on network conditions và task requirements
4. **Continual learning:** Models adapt to concept drift và new anomaly types without forgetting
5. **Incentive mechanisms:** Economic models để khuyến khích participation trong DFL networks

#### **Research directions:**

- **Theory:** Convergence guarantees cho DFL với Non-IID data và dynamic topologies
- **Optimization:** Communication-efficient algorithms minimize rounds to convergence
- **Fairness:** Đảm bảo clients với ít data vẫn benefit từ global model
- **Explainability:** Interpret models learned từ distributed data

## **6.4 Tác Động Thực Tiễn**

DFL có tiềm năng transform các ngành công nghiệp:

- **Manufacturing:** Giảm downtime thông qua predictive maintenance với privacy
- **Healthcare:** Hospitals collaborate trên medical diagnosis mà không share patient data
- **Transportation:** Autonomous vehicles learn driving policies từ fleet without central data collection

- **Energy:** Smart grids optimize operations collaboratively

#### Societal impact:

- Empowers individuals với data ownership và privacy
- Enables AI for organizations không đủ resources cho centralized infrastructure
- Supports regulatory compliance (GDPR, HIPAA)
- Promotes democratization of AI

## Tài Liệu Tham Khảo

1. McMahan, B., Moore, E., Ramage, D., Hampson, S., & y Arcas, B. A. (2017). Communication-efficient learning of deep networks from decentralized data. *Artificial Intelligence and Statistics*, 1273-1282.
2. Kairouz, P., McMahan, H. B., et al. (2021). Advances and open problems in federated learning. *Foundations and Trends in Machine Learning*, 14(1-2), 1-210.
3. Lim, W. Y. B., Luong, N. C., et al. (2020). Federated learning in mobile edge networks: A comprehensive survey. *IEEE Communications Surveys & Tutorials*, 22(3), 2031-2063.
4. Roy, A. G., Siddiqui, S., et al. (2019). BrainTorrent: A peer-to-peer environment for decentralized federated learning. *arXiv preprint arXiv:1905.06731*.
5. Lalitha, A., Shekhar, S., Javidi, T., & Koushanfar, F. (2019). Fully decentralized federated learning. *Third Workshop on Machine Learning on the Phone and other Consumer Devices*.
6. Beutel, D. J., Topal, T., et al. (2020). Flower: A friendly federated learning research framework. *arXiv preprint arXiv:2007.14390*.
7. Li, T., Sahu, A. K., Zaheer, M., Sanjabi, M., Talwalkar, A., & Smith, V. (2020). Federated optimization in heterogeneous networks. *Proceedings of Machine Learning and Systems*, 2, 429-450.
8. NASA. (2007). Bearing Dataset. IMS, University of Cincinnati. <https://ti.arc.nasa.gov/tech/dash/groups/pcoe/prognostic-data-repository/>