

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC BÁCH KHOA



**CROSS-DEVICE
DECENTRALIZED FEDERATED LEARNING**

Giảng viên: TS. Phan Trọng Nhân
Học viên: Nguyễn Văn Tâm - 2570358
 Đồng Quang Trí - 2570523
 Phạm Văn Thành - 2570496
 Lê Đức Phương - 2570480
 Dinh Thị Thu Thủy - 2570508

Thành viên và nhiệm vụ

No.	Họ và tên	Nhiệm vụ	MSSV	Phần trăm hoàn thành
1	Nguyễn Văn Tâm	Chương 1 - 2	2570358	20 %
2	Phạm Văn Thành	Chương 3	2570496	20 %
3	Lê Đức Phương	Chương 4	2570480	20 %
4	Đồng Quang Trí	Chương 5	2570523	20 %
5	Đinh Thị Thu Thủy	Chương 6	2570508	20 %

Danh sách từ viết tắt

Viết tắt	Giải thích
IoT	Internet of Things
FL	Federated Learning
DFL	Decentralized Federated Learning
IID	Independent and Identically Distributed
Non-IID	Non-Independent and Identically Distributed
P2P	Peer-to-Peer
WSN	Wireless Sensor Network
IIoT	Industrial Internet of Things
GDPR	General Data Protection Regulation
HIPAA	Health Insurance Portability and Accountability Act
FedAvg	Federated Averaging
FedProx	Federated Proximal
FedOpt	Federated Optimization
MSE	Mean Squared Error

Mục lục

Chương 1

Giới thiệu

1.1 Đặc trưng của dữ liệu IoT

Hệ thống IoT sở hữu nhiều đặc điểm nổi bật, giúp nó trở thành một công nghệ đột phá và thay đổi cách chúng ta tương tác với thế giới xung quanh. Trong đó có các đặc điểm chính được trình bày dưới đây:

- Khối lượng (Volume): Các thiết bị IoT tạo ra lượng dữ liệu khổng lồ, dễ gây quá tải cho hệ thống lưu trữ và xử lý.
- Tốc độ (Velocity): Dữ liệu thường được phát sinh theo thời gian thực, đòi hỏi khả năng xử lý và phân tích nhanh chóng để đảm bảo tính kịp thời.
- Đa dạng (Variety): Dữ liệu IoT tồn tại ở nhiều dạng khác nhau, từ có cấu trúc đến phi cấu trúc, yêu cầu các phương pháp phân tích linh hoạt và thích ứng.
- Nhiều và không đầy đủ (Low Veracity): Cảm biến dễ sai số, mất mẫu hoặc nhiều dữ liệu.

Việc quản lý và khai thác hiệu quả dữ liệu IoT đòi hỏi các giải pháp công nghệ tiên tiến, bao gồm hệ thống lưu trữ phân tán, công cụ phân tích thời gian thực và các thuật toán xử lý dữ liệu đa dạng.

1.2 Bối cảnh dữ liệu phân tán trong IoT

Sự phát triển mạnh mẽ của Internet of Things (IoT) đang tạo ra một hệ sinh thái với hàng chục tỷ thiết bị thông minh hoạt động liên tục trong nhiều lĩnh vực như sản xuất công nghiệp, đô thị thông minh, năng lượng và giao thông. Dự báo cho thấy số lượng thiết bị IoT toàn cầu có thể vượt hơn 75 tỷ vào năm 2025. Các thiết bị này thu thập dữ liệu theo thời gian thực từ nhiều loại cảm biến khác nhau, bao gồm cảm biến rung, nhiệt độ, âm thanh, áp suất và các thiết bị đo lường khác.

Dữ liệu IoT có tính phân tán tự nhiên vì các thiết bị:

- Được triển khai rộng rãi về mặt địa lý;
- Tạo ra dữ liệu liên tục với tần suất cao;
- Có sự không đồng nhất về chất lượng và đặc trưng (Non-IID);
- Chịu ảnh hưởng bởi điều kiện môi trường và tải trọng khác nhau.

Việc thu thập và xử lý dữ liệu IoT theo phương pháp truyền thống, trong đó dữ liệu được tải về một máy chủ trung tâm, trở nên không khả thi do hạn chế về băng thông, độ trễ và yêu cầu bảo mật. Điều này đặt ra nhu cầu về các mô hình học máy phân tán, phù hợp với tính chất của dữ liệu IoT.

1.3 Vấn đề bảo mật, băng thông và tính sẵn sàng

Hệ thống IoT hiện nay phải đối mặt với ba nhóm thách thức chính:

Bảo mật và quyền riêng tư: Dữ liệu cảm biến trong môi trường công nghiệp thường chứa thông tin nhạy cảm như tình trạng máy móc, quy trình sản xuất và thông số vận hành. Việc truyền dữ liệu thô về server trung tâm tiềm ẩn rủi ro rò rỉ thông tin, tấn công mạng và vi phạm các quy định như GDPR hoặc HIPAA.

Giới hạn băng thông: Dữ liệu IoT, đặc biệt dạng tín hiệu rung có tần số lấp mău cao (ví dụ 20 kHz), tạo ra lượng dữ liệu khổng lồ. Việc truyền liên tục dữ liệu thô từ hàng trăm thiết bị gây quá tải mạng, làm tăng chi phí vận hành và không phù hợp với các môi trường có kết nối yếu.

Độ trễ và tính sẵn sàng: Các ứng dụng như giám sát thiết bị theo thời gian thực hoặc bảo trì dự đoán (predictive maintenance) yêu cầu độ trễ thấp và tính sẵn sàng cao. Việc phụ thuộc vào cloud khiến hệ thống dễ bị gián đoạn khi mất mạng, đồng thời làm tăng độ trễ trong xử lý.

Những hạn chế này đòi hỏi các cơ chế học tập phân tán hiệu quả và an toàn hơn.

1.4 Tại sao cần Decentralized Federated Learning cho IoT

Federated Learning (FL) truyền thống giải quyết một phần vấn đề bằng cách cho phép các thiết bị IoT huấn luyện mô hình cục bộ và chỉ gửi tham số mô hình (model updates) lên server trung tâm. Tuy nhiên, FL vẫn mang cấu trúc tập trung vì:

- Server trung tâm là điểm duy nhất để tổng hợp tham số;
- Nếu server gặp sự cố, toàn bộ hệ thống ngừng hoạt động;
- Tất cả các thiết bị đều phải giao tiếp với server, tạo ra nút thắt cổ chai (bottleneck);
- Server trở thành mục tiêu tấn công và là nơi tập trung rủi ro.

Decentralized Federated Learning (DFL) ra đời để loại bỏ hoàn toàn sự phụ thuộc vào server trung tâm. Trong DFL, các thiết bị IoT giao tiếp trực tiếp với nhau theo mô hình peer-to-peer (P2P) thông qua các topology như ring, mesh hoặc gossip. Điều này mang lại:

- Độ trễ thấp do giao tiếp nội bộ (local communication);
- Khả năng chịu lỗi cao vì mạng vẫn hoạt động ngay cả khi một số thiết bị bị ngắt kết nối;
- Tính mở rộng tốt hơn so với FL truyền thống;
- Phù hợp với mô hình edge computing và các hệ thống IoT không kết nối cloud.

Với những đặc tính này, DFL đặc biệt phù hợp cho bài toán học máy trên IoT phân tán và không đồng nhất.

1.5 Mục tiêu và phạm vi báo cáo

Báo cáo hướng tới việc nghiên cứu và đánh giá tiềm năng của Decentralized Federated Learning trong bối cảnh hệ thống IoT. Các mục tiêu chính bao gồm:

- Trình bày tổng quan về Federated Learning và Decentralized Federated Learning.
- Phân tích các thách thức trong học máy phân tán trên IoT và lý do DFL là cần thiết.
- Triển khai mô phỏng mô hình DFL với 10 thiết bị IoT trong kiến trúc P2P ring topology.
- Áp dụng mô hình Autoencoder cho bài toán phát hiện bất thường cảm biến vòng bi.
- So sánh hiệu suất giữa phân phối dữ liệu cân bằng (IID) và không cân bằng (Non-IID).
- Dánh giá ưu, nhược điểm và đề xuất hướng phát triển.

Phạm vi báo cáo tập trung vào mô phỏng trong môi trường phần mềm, sử dụng bộ dữ liệu rung vòng bi NASA và triển khai thuận túy DFL với kiến trúc P2P (Peer-to-Peer) ring topology. Báo cáo không đi sâu vào các cơ chế bảo mật nâng cao hoặc triển khai trên phần cứng thực tế.

Chương 2

Decentralized Federated Learning (DFL)

2.1 Khái niệm FL và DFL

Federated Learning (FL) [?] là phương pháp học máy phân tán trong đó mô hình được huấn luyện trực tiếp trên từng thiết bị hoặc node mà không cần tập trung dữ liệu về server trung tâm. Các thiết bị chỉ gửi *model updates* (như gradients hoặc weights) thay vì gửi dữ liệu thô, giúp giảm rủi ro rò rỉ thông tin và giảm tải băng thông. Tuy nhiên, FL truyền thống vẫn phụ thuộc vào một server trung tâm để tổng hợp mô hình, tạo ra điểm nghẽn (bottleneck) và rủi ro *single point of failure*.

Trong khi đó, Decentralized Federated Learning (DFL) loại bỏ hoàn toàn sự phụ thuộc vào server tổng hợp. Các thiết bị IoT giao tiếp trực tiếp theo mô hình *peer-to-peer*, thực hiện trao đổi mô hình và cập nhật đồng thuận theo cấu trúc mạng được định nghĩa trước (ring, mesh, gossip, hoặc blockchain-based).

Mục tiêu chính của DFL:

- Loại bỏ điểm tập trung, tăng khả năng chịu lỗi và bảo mật.
- Tận dụng giao tiếp cục bộ (local communication) để giảm độ trễ và chi phí mạng.
- Hỗ trợ các hệ thống IoT quy mô lớn, phân tán và không đồng nhất.
- Đảm bảo tính riêng tư và tính tự chủ của từng thiết bị.

2.2 Kiến trúc DFL

Khác với FL truyền thống dựa vào một server trung tâm, kiến trúc DFL được xây dựng hoàn toàn theo mô hình mạng phân tán. Một số topology phổ biến:

Peer-to-Peer (P2P): Mỗi node kết nối với một số node lân cận, thực hiện trao đổi mô hình trực tiếp. P2P đơn giản, linh hoạt, phù hợp IoT và không yêu cầu quản lý tập trung.

Ring Topology: Các node được sắp xếp thành một vòng khép kín. Mỗi node chỉ giao tiếp với hai node liền kề. Topology này dễ triển khai, chi phí giao tiếp thấp nhưng tốc độ hội tụ chậm.

Gossip-Based Topology: Mỗi node ngẫu nhiên chọn một hoặc nhiều neighbor để chia sẻ model weights. Đây là cấu trúc phổ biến trong DFL, đảm bảo hội tụ nhanh, phân phối tải đều, và có khả năng mở rộng tốt.

Blockchain-Based DFL: Sử dụng blockchain để ghi lại và xác thực các bản cập nhật mô hình. Topology này đảm bảo tính toàn vẹn và chống gian lận, nhưng chi phí tính toán và độ trễ cao hơn.

Nhìn chung, các kiến trúc trên đều hướng tới mục tiêu: không có server trung tâm, giao tiếp phi tập trung, và tối ưu cho các hệ thống IoT với hạ tầng mạng linh hoạt.

2.3 Quy trình hoạt động của DFL

Quy trình trong DFL diễn ra hoàn toàn theo dạng phân tán mà không cần server tổng hợp. Một vòng lặp tiêu biểu của DFL bao gồm:

1. **Khởi tạo mô hình** tại mỗi node hoặc nhận mô hình từ một node khởi tạo.
2. **Local training**: mỗi node huấn luyện mô hình dựa trên dữ liệu của riêng mình.
3. **Peer selection**: node chọn một hoặc nhiều node lân cận theo topology.
4. **Exchange**: các node trao đổi weights/model updates theo dạng P2P.
5. **Aggregation**: node cập nhật mô hình bằng cách trung bình hoá (average hoặc weighted average) các mô hình từ neighbors.
6. **Update**: mô hình mới được lưu lại tại node và tiếp tục training.
7. **Lặp lại** cho đến khi mô hình hội tụ.

Không có node nào giữ vai trò trung tâm, và hệ thống vẫn tiếp tục hoạt động khi một số node bị lỗi hoặc tách khỏi mạng.

2.4 Xử lý trường hợp lỗi node hoặc kết nối không ổn định trong DFL

Trong môi trường IoT, sự cố node và kết nối không ổn định là vấn đề rất phổ biến. Thiết bị IoT thường có tài nguyên hạn chế, năng lượng thấp, hoạt động theo chu kỳ ngủ hoặc thức, đồng thời phải truyền dữ liệu qua mạng không dây nhiều lần. Do đó, một mô hình DFL hiệu quả phải tích hợp các cơ chế để đảm bảo việc huấn luyện phân tán vẫn diễn ra ổn định và mô hình vẫn hội tụ ngay cả khi có lỗi.

1. Cơ chế truyền thông bất đồng bộ (Asynchronous Communication)

- Mỗi node có thể gửi hoặc nhận cập nhật mô hình vào thời điểm khác nhau.
- Node nhanh không phải chờ node chậm (straggler).

- Mạng không bị “đứng” khi có node mất kết nối tạm thời.

2. Linh hoạt trong kiến trúc liên lạc (Temporal Variability): DFL có thể:

- Thay đổi tuyến truyền mô hình.
- Loại bỏ tạm thời node mất ổn định.
- Kết nối lại khi node hoạt động trở lại.

Nhờ vậy mô hình vẫn hội tụ ngay cả khi topology thay đổi liên tục.

3. Thuật toán Gossip giúp chống lỗi tự nhiên (Gossip Robustness)

- Mỗi node chỉ cần trao đổi với một nhóm nhỏ các node lân cận..
- Một node chết không làm “đứt” toàn bộ vòng huấn luyện.
- Mô hình vẫn lan truyền qua các đường khác trong mạng.

Đây là cơ chế đặc biệt phù hợp với mạng IoT mesh và IoT công nghiệp.

4. Cơ chế bỏ qua hoặc giảm ảnh hưởng của Stragglers

- Các node không cần chờ nhau.
- Node chậm sẽ không làm ngưng toàn bộ tiến trình.
- Khi nhận mô hình quá cũ, node nhanh có thể bỏ qua cập nhật lạc hậu (stale update).

Điều này giúp mô hình hội tụ nhanh hơn và ổn định hơn so với FL tập trung.

5. Cơ chế chống lỗi truyền (Noise/ Packet Loss Handling)

- Truyền nhiều bản sao mô hình trong broadcast-gossip.
- Truyền mô hình nhỏ gọn hơn (gradient compression).
- Lọc hoặc làm mượt mô hình bị nhiễu thông qua averaging.

Điều này đặc biệt hữu ích trong IoT vì môi trường mạng không dây dễ nhiễu.

6. Node lỗi không làm gián đoạn quá trình hội tụ: Do không có server trung tâm, DFL không bị phụ thuộc vào một node duy nhất. Điều này giải quyết hai rủi ro lớn:

- Không có “single point of failure: Một node chết thì chỉ mất một nguồn dữ liệu, quá trình học vẫn tiếp tục.

- Không tắc nghẽn tại server: Không có node đóng vai trò bottleneck của toàn bộ mạng.
7. **Kết hợp Blockchain để chống gian lận và lỗi mô hình:** Một số hệ thống DFL sử dụng blockchain để bảo đảm tính toàn vẹn mô hình, phát hiện node gửi mô hình sai hoặc tấn công độc hại và chống giả mạo cập nhật. Mặc dù không bắt buộc, nhưng với IoT quy mô lớn, blockchain giúp tăng độ tin cậy.

2.5 Ưu điểm của DFL so với FL trong IoT

DFL sở hữu nhiều ưu điểm quan trọng đối với các hệ thống IoT:

- **Không có single point of failure:** mạng hoạt động ngay cả khi nhiều node ngừng hoạt động.
- **Độ trễ thấp:** giao tiếp chủ yếu diễn ra ở local network, thay vì gửi lên cloud.
- **Khả năng mở rộng cao:** số lượng node lớn không tạo ra nghẽn mạng như mô hình FL truyền thống.
- **Tính riêng tư mạnh hơn:** không có trung tâm lưu trữ model updates.
- **Phù hợp môi trường edge:** hoạt động tốt khi kết nối không ổn định hoặc gián đoạn.
- **Tận dụng cluster tự nhiên của IoT:** các sensors trong cùng khu vực có thể liên kết với nhau hiệu quả.

Những đặc điểm này khiến DFL trở thành kiến trúc lý tưởng cho các ứng dụng như smart factory, smart home, autonomous vehicles và mạng cảm biến phân tán.

2.6 Thách thức kỹ thuật của DFL

Mặc dù mang lại nhiều lợi thế, DFL vẫn tồn tại các thách thức đáng kể:

Hội tụ khó khăn: Do không có server trung tâm, mô hình toàn hệ thống dễ bị phân kỳ, đặc biệt trong môi trường Non-IID [?] và khi các updates không đồng bộ.

Communication overhead: Mỗi node phải trao đổi mô hình với nhiều neighbors, dẫn đến tổng chi phí giao tiếp cao hơn FL truyền thống [?].

Bảo mật và tính toàn vẹn: Mạng P2P dễ bị tấn công như:

- Byzantine attacks

- Model poisoning
- Backdoor injection
- Inference attacks

Không có server trung tâm để kiểm tra tính đúng đắn của model updates.

Thiết kế topology tối ưu: Cần cân bằng giữa tốc độ hội tụ, chi phí truyền thông và khả năng chịu lỗi.

Heterogeneity: IoT có sự khác biệt lớn về:

- kích thước dữ liệu,
- năng lực tính toán,
- trạng thái kết nối,
- tiêu thụ năng lượng.

DFL cần điều chỉnh phù hợp với môi trường đa dạng này.

Tóm lại, DFL mang lại nền tảng vững chắc để triển khai học máy phân tán trong IoT, nhưng yêu cầu nghiên cứu thêm về bảo mật, tối ưu topology và đảm bảo hội tụ.

2.7 Ứng Dụng DFL Trong Hệ Thống IoT Thực Tế

1. **Smart City:** 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 chuyển dữ liệu quan trọng về máy chủ tập trung.
- Dự đoán và cảnh báo theo thời gian thực.

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

- Quyền riêng tư: Không chia sẻ thói quen sinh hoạt với bất kỳ bên thứ ba nào.
- Cá nhân hóa: Mô hình cục bộ được tối ưu để phù hợp với từng hộ gia đình.
- Hợp tác: Học hỏi từ các thiết bị lân cận mà không tiết lộ dữ liệu cá nhân.

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

- Các trạm biến áp huấn luyện mô hình dựa trên mẫu tiêu thụ điện năng tại chỗ.
- Chia sẻ mô hình với các trạm lân cận để nâng cao độ chính xác dự đoán.
- Phát hiện bất thường như hành vi trộm điện hoặc sự cố thiết bị.

4. **Industrial IoT (IIoT):** Trường hợp sử dụng chính của báo cáo này: Predictive maintenance trong nhà máy.

- Mỗi máy hoặc dây chuyền sản xuất là một node.
- Các node trong cùng một nhà máy tạo thành một cụm với kết nối nội bộ nhanh.
- Các cụm từ các nhà máy khác nhau có thể liên kết với nhau ở tốc độ chậm hơn.

5. **Wireless Sensor Networks (WSN):** Mạng cảm biến không dây triển khai trong môi trường từ xa (rừng, đại dương, nông nghiệp):

- Các cảm biến giao tiếp với nút lân cận khi nằm trong phạm vi sóng vô tuyến.
- Cập nhật mô hình lan truyền qua giao thức gossip.
- Tiết kiệm năng lượng: chỉ chia sẻ mô hình khi có đủ dữ liệu mới.
- Chịu lỗi: mạng tự phục hồi khi cảm biến gặp sự cố.

Chương 3

Mô phỏng DFL cho IoT (không dùng thiết bị thật)

3.1 Lý do và yêu cầu mô phỏng

Trong bối cảnh IoT, việc triển khai các thuật toán Distributed Federated Learning (DFL) trực tiếp trên thiết bị thật gặp nhiều hạn chế:

- Chi phí phần cứng cao: IoT nodes có số lượng lớn, việc chuẩn bị 10–50 thiết bị để thí nghiệm là không khả thi.
- Tài nguyên phần cứng hạn chế: Nhiều thiết bị thực (cảm biến, vi điều khiển, gateway nhỏ) không đủ năng lực để huấn luyện mô hình phức tạp.
- Mạng IoT khó tái tạo: Các yếu tố như mốc gói, độ trễ cao, hoặc thay đổi topology rất khó mô phỏng chính xác trên thiết bị thật.

Do đó, mô phỏng DFL cho phép nghiên cứu hành vi của hệ thống mà không cần triển khai vật lý, mang lại các lợi ích:

- Kiểm soát hoàn toàn môi trường (số node, băng thông, độ trễ, lỗi kết nối).
- Tiết kiệm chi phí và thời gian.
- Lặp lại (reproducibility) dễ dàng.
- Hỗ trợ thử nghiệm nhiều topology (mesh, ring, random gossip) mà không cần thay đổi phần cứng.

Việc mô phỏng đặc biệt quan trọng trong IoT vì môi trường mạng không ổn định và thiết bị yếu đều là các yếu tố mà DFL cần phải xử lý.

3.2 Công cụ mô phỏng

Trong nghiên cứu này, nhóm triển khai DFL thuần túy với kiến trúc P2P (Peer-to-Peer) tùy chỉnh:

- **Kiến trúc P2P Ring Topology**
 - Mỗi peer kết nối với 2 peer lân cận trong mô hình vòng tròn.
 - Mô hình được trao đổi theo chiều kim đồng hồ giữa các peer.
 - Không có central server - hoàn toàn phi tập trung (serverless).
 - Mỗi peer thực hiện aggregation cục bộ với mô hình nhận được từ peer trước đó.

- **Triển khai bằng Python**

- Sử dụng PyTorch cho training và model management.
- Mỗi peer được mô phỏng như một process độc lập.
- Communication giữa các peer được mô phỏng thông qua trao đổi trọng số mô hình.
- Hỗ trợ cả phân phối dữ liệu IID (balanced) và Non-IID (imbalanced).
- Lấy cảm hứng từ Flower framework [?] nhưng triển khai thuận túy DFL P2P.

- **Ưu điểm của phương pháp này**

- Hoàn toàn phi tập trung - không phụ thuộc vào server trung tâm.
- Tăng tính riêng tư và bảo mật dữ liệu.
- Giảm single point of failure.
- Phù hợp cho môi trường IoT với tài nguyên hạn chế.
- Dễ dàng mở rộng và tùy chỉnh topology.

3.3 Định nghĩa bài toán mô phỏng

Trong mô phỏng DFL cho IoT, nhóm sử dụng bài toán phát hiện bất thường (anomaly detection) trong máy quay công nghiệp dựa trên dữ liệu rung (vibration). Đây là một bài toán điển hình trong IoT công nghiệp (IIoT), nơi cảm biến được lắp trực tiếp lên các vòng bi, mô tơ hoặc robot để phát hiện lỗi trước khi sự cố xảy ra.

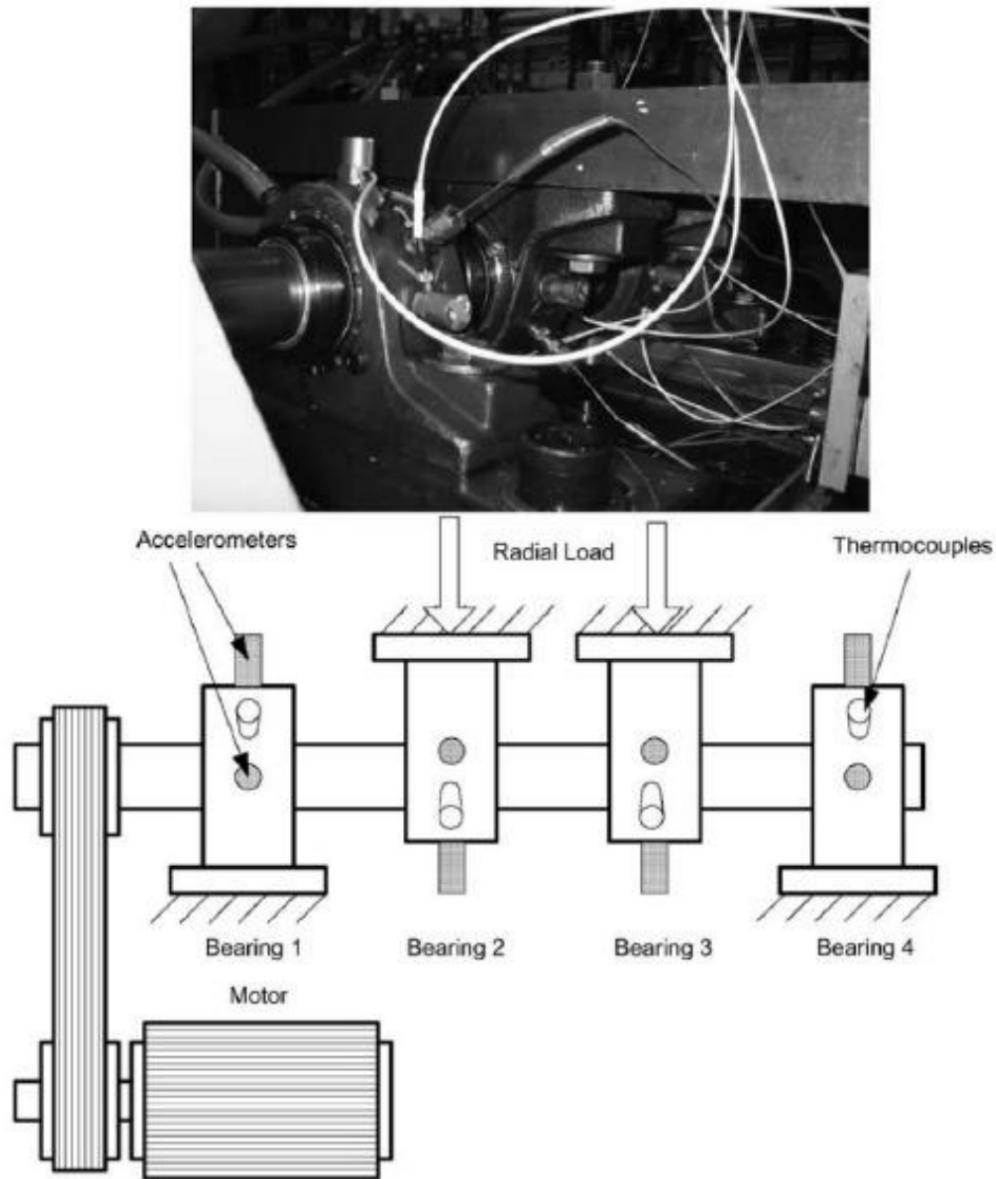
- Nhiệm vụ: Mỗi thiết bị IoT (mỗi peer) học mô hình phát hiện bất thường dựa trên dữ liệu rung của vòng bi.
- Mục tiêu: Mô phỏng cách các cảm biến IoT, mỗi thiết bị chứa lượng dữ liệu khác nhau, cùng học một mô hình phân tán mà không cần chia sẻ dữ liệu thô.
- Kịch bản IoT:
 - Mỗi peer tương ứng với một cảm biến gắn trên máy.
 - Mỗi cảm biến ghi nhận dữ liệu theo thời gian, tạo ra dataset riêng (IID hoặc Non-IID).
 - Các cảm biến trao đổi trọng số mô hình thông qua cơ chế DFL P2P ring topology.

Chương 4

Triển khai mô hình DFL

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.



Hình 4.1: Thiết bị trong công nghiệp

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.

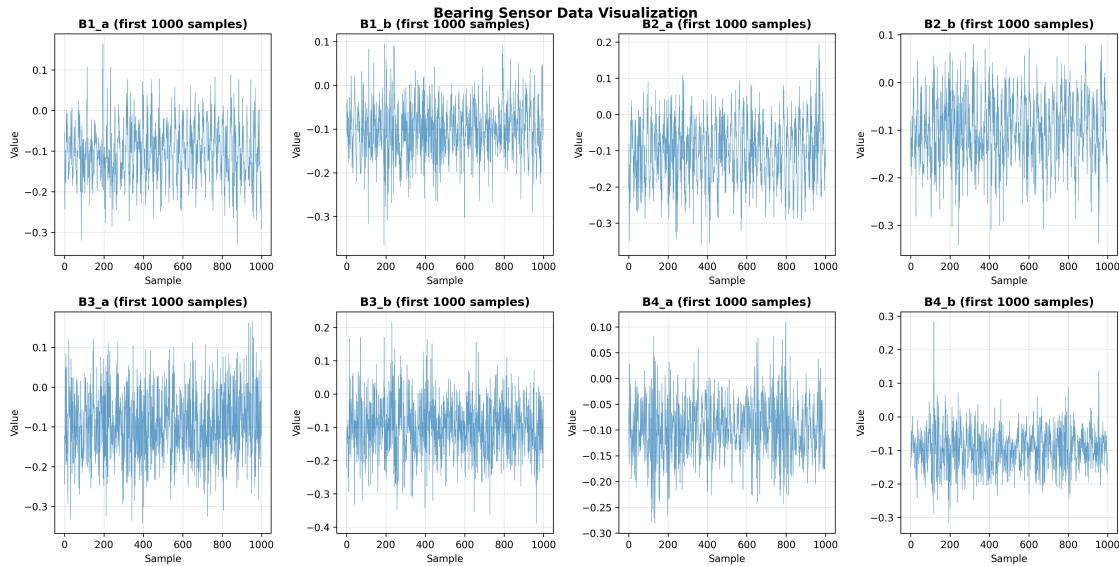
Imbalanced (Non-IID): Phân phối theo power law để mô phỏng thực tế:

- Client 0: 9,830 samples (100.0%)
- Client 1: 1,638 samples (14.3%)
- Client 2: 3,932 samples (25.5%)
- Client 3: 3,276 samples (17.5%)
- Client 4: 2,948 samples (13.6%)
- Client 5: 2,620 samples (10.8%)
- Client 6: 2,620 samples (9.8%)
- Client 7: 2,293 samples (7.9%)
- Client 8: 3,276 samples (10.1%)
- Client 9: 329 samples (1.0%)

Bảng 4.1: Phân phối dữ liệu giữa 10 clients

Client	Train (IID)	Train (Non-IID)	Test (IID)	Test (Non-IID)
0	3,276	9,830 (100.0%)	820	2,458
1	3,276	1,638 (14.3%)	820	410
2	3,276	3,932 (25.5%)	820	983
3	3,276	3,276 (17.5%)	820	820
4	3,276	2,948 (13.6%)	820	738
5	3,276	2,620 (10.8%)	820	656
6	3,276	2,620 (9.8%)	820	656
7	3,276	2,293 (7.9%)	820	574
8	3,276	3,276 (10.1%)	820	820
9	3,276	329 (1.0%)	820	83

4.2 Sensor Data Visualization



Hình 4.2: 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

4.3 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.

MSE Formula và Computation:

Loss function được sử dụng là Mean Squared Error (MSE):

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

Trong đó:

- x_i : giá trị gốc từ cảm biến thứ i
- \hat{x}_i : giá trị tái tạo từ autoencoder cho cảm biến thứ i
- n : số lượng features (8 sensors trong bearing dataset)

Decision Rule cho Anomaly Detection:

$$\text{Classification} = \begin{cases} \text{NORMAL} & \text{if } MSE < \text{Threshold} \\ \text{ANOMALY} & \text{if } MSE \geq \text{Threshold} \end{cases}$$

Threshold Calculation:

$$\text{Threshold} = \text{Percentile}_{95}(\{MSE_i\}_{i=1}^N)$$

Trong đó N là tổng số test samples trong validation set.

4.4 Cấu Hình Decentralized Federated Learning

Triển khai DFL thuận túy với kiến trúc P2P Ring Topology:

- **Số peers:** 10 (mô phỏng 10 thiết bị IoT)
- **Topology:** Ring - mỗi peer kết nối với 2 peer lân cận
- **Communication:** Peer-to-Peer (P2P) - không có central server
- **Aggregation:** Local tại mỗi peer (trung bình trọng số từ peer trước đó)
- **Total rounds:** 50
- **Local epochs:** 1 epoch mỗi round
- **Batch size:** 128
- **Learning rate:** 0.001
- **Device:** CPU

Ring Topology Flow:

- Peer 0 → Peer 1 → Peer 2 → ... → Peer 9 → Peer 0
- Mỗi peer train local model và gửi weights cho peer tiếp theo
- Peer nhận weights từ peer trước đó và thực hiện aggregation (averaging)
- Không có single point of failure - hoàn toàn phi tập trung

Ưu điểm của DFL P2P:

- Tăng tính riêng tư: dữ liệu không bao giờ rời khỏi peer
- Giảm bandwidth: chỉ giao tiếp với 2 peer lân cận
- Fault tolerance: nếu một peer fail, topology có thể tái cấu hình
- Scalable: dễ dàng thêm/bớt peer mà không ảnh hưởng toàn hệ thống

4.5 Quy Trình Thực Nghiệm

Experiment 1: Balanced (IID)

1. Chia đều 32,760 samples cho 10 peers (3,276 samples/peer)
2. Khởi tạo model với random weights
3. Chạy 50 rounds DFL với P2P ring topology
4. Mỗi peer train local model và trao đổi weights với peer kế tiếp
5. Đánh giá trên local test set của mỗi peer

Experiment 2: Imbalanced (Non-IID)

1. Phân phối không cân bằng theo power law (từ 329 đến 9,830 samples)
2. Các bước còn lại giống Experiment 1

Metrics:

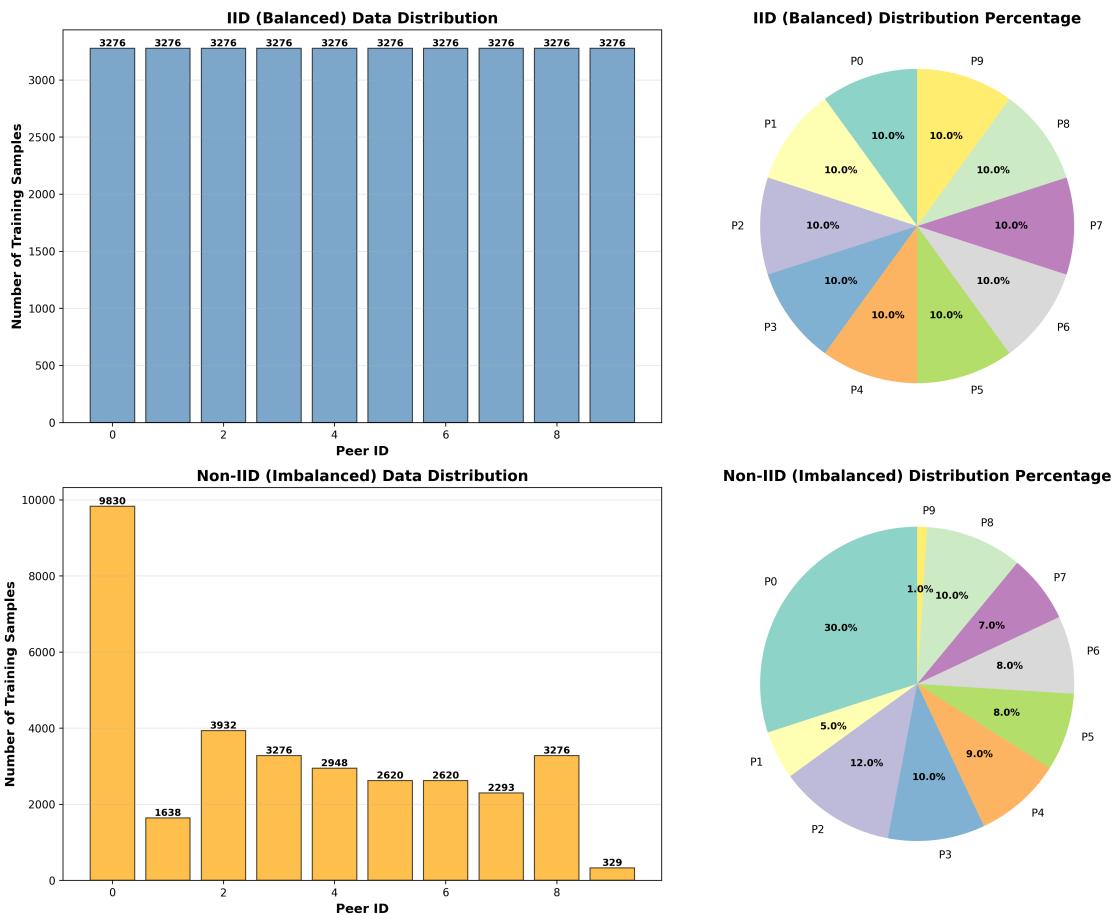
- Training loss: MSE trên training set của mỗi peer
- Evaluation loss: MSE trên test set của mỗi peer
- Convergence speed: số rounds để đạt $\text{loss} < 0.005$
- Loss reduction: chênh lệch giữa initial loss và final loss
- Anomaly detection threshold: 95th percentile và Mean+ 2σ của MSE distribution

Chương 5

Kết quả mô phỏng và Đánh giá

5.1 Dataset và Phân Phối Dữ Liệu Giữa Các Peers

Trong kiến trúc Decentralized Federated Learning (DFL), dữ liệu được phân phối giữa 10 peers theo cấu trúc ring topology. Chúng tôi thực hiện hai scenarios phân phối dữ liệu:



Hình 5.1: So sánh phân phối dữ liệu Balanced vs Imbalanced trong DFL

Phân tích phân phối dữ liệu:

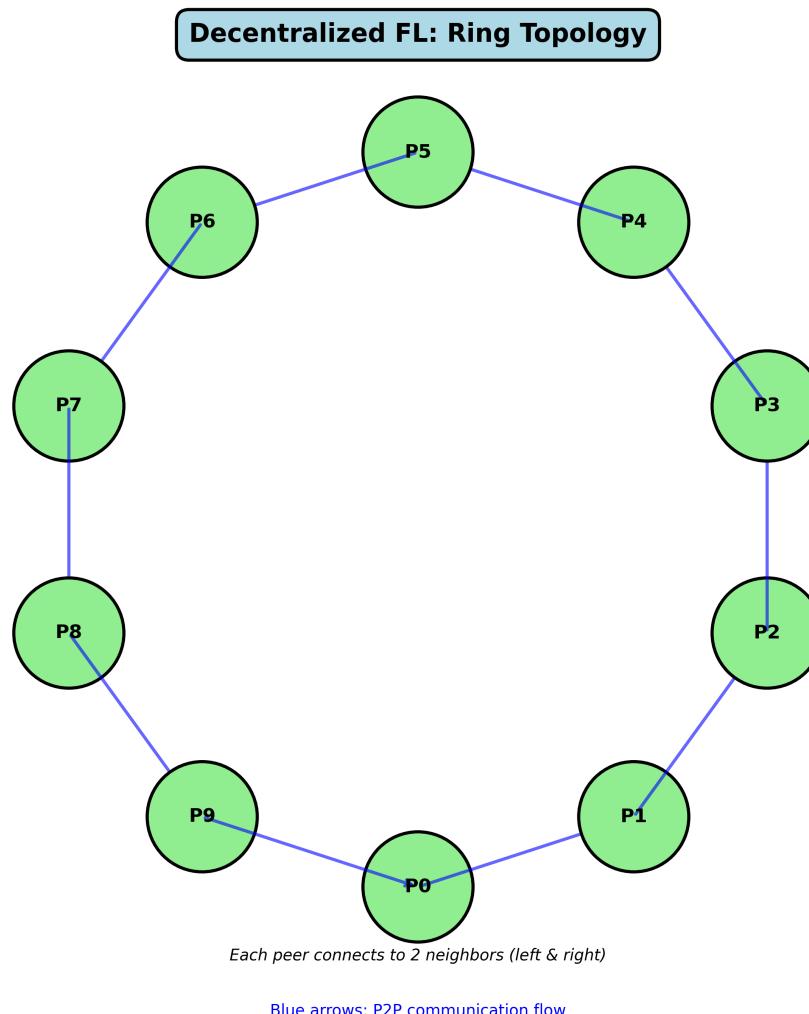
Bảng 5.1: Thống kê phân phối dữ liệu

Distribution	Total Samples	Mean	Std Dev	Range
Balanced	32,760	3,276.0	0.0	3,276 - 3,276
Imbalanced	32,762	3,276.2	2,380.9	329 - 9,830

Quan sát từ visualization:

- **Balanced Distribution:** Mỗi peer có số lượng samples đồng đều (3,276 samples), tạo ra phân phối cân bằng hoàn hảo với 10% data cho mỗi peer
- **Imbalanced Distribution:** Peer 0 chiếm 30% tổng data (9,830 samples), trong khi Peer 9 chỉ có 1% (329 samples) - chênh lệch gấp 30 lần
- Standard deviation của imbalanced distribution (2,380.9) cho thấy độ biến thiên cao
- Bar charts minh họa sự chênh lệch đáng kể về số lượng samples giữa các peers trong scenario imbalanced

5.2 Kiến Trúc Decentralized Federated Learning



Hình 5.2: Sơ đồ kiến trúc Ring Topology trong Decentralized Federated Learning

Đặc điểm kiến trúc DFL:

- **Peer-to-Peer Communication:** Không có central server, các peers giao tiếp trực tiếp với nhau theo cấu trúc ring
- **Ring Topology:** Mỗi peer chỉ kết nối với peer kế tiếp, tạo thành một vòng khép kín
- **Local Aggregation:** Mỗi peer tự thực hiện aggregation với model từ peer trước đó

- **Model Architecture:** Autoencoder với cấu trúc Input(8) → Hidden(16) → Latent(4) → Hidden(16) → Output(8)
- **Sequential Update:** Model weights được truyền tuần tự từ Peer 0 → Peer 1 → ... → Peer 9 → Peer 0

Training Process trong DFL:

1. Mỗi peer train model trên local data
2. Peer nhận model từ peer trước đó
3. Thực hiện averaging: $w_{new} = \alpha \cdot w_{local} + (1 - \alpha) \cdot w_{received}$
4. Gửi model đã aggregate đến peer tiếp theo
5. Sau khi hoàn thành một vòng, bắt đầu round mới

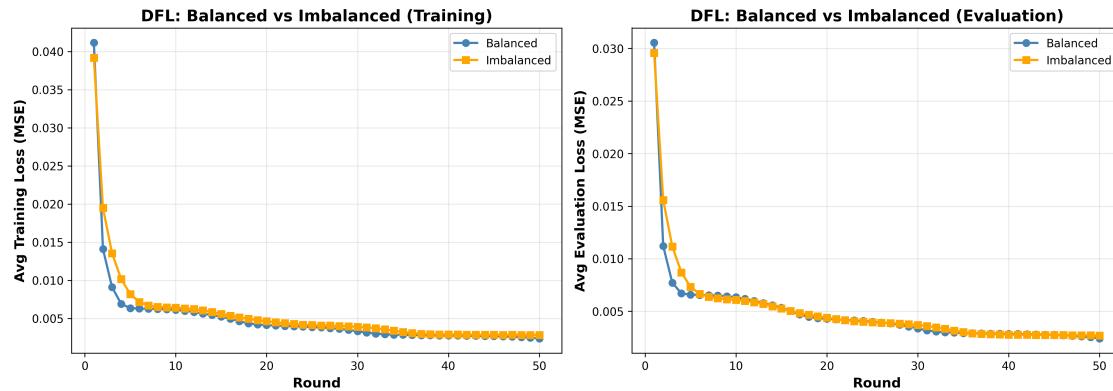
Configuration:

- Number of Peers: 10
- Training Rounds: 50
- Local Epochs: 1
- Learning Rate: 0.001
- Batch Size: 128
- Device: CPU

5.3 Hiệu Suất Training

Bảng 5.2: Kết quả thí nghiệm DFL

Experiment	Initial Loss	Final Train Loss	Final Eval Loss	Reduction
DFL Balanced	0.041149	0.002392	0.002425	94.19%
DFL Imbalanced	0.039153	0.002842	0.002705	92.74%



Hình 5.3: So sánh quá trình training của hai thí nghiệm DFL

Quan sát từ kết quả:

- **Balanced Distribution** cho kết quả tốt hơn với final eval loss (0.002425) thấp hơn Imbalanced (0.002705)
- Train loss reduction của Balanced (94.19%) cao hơn đáng kể so với Imbalanced (92.74%)
- Cả hai experiments đều hội tụ ổn định sau khoảng 30-40 rounds
- Balanced distribution có learning curve mượt mà hơn, ít fluctuation hơn

Phân tích chi tiết:

1. DFL Balanced:

- Convergence nhanh và ổn định nhờ phân phối đều dữ liệu
- Mỗi peer đóng góp đều nhau vào việc cập nhật model
- Giảm bias từ peers có nhiều data
- Final loss thấp nhất: 0.002392 (train), 0.002425 (eval)

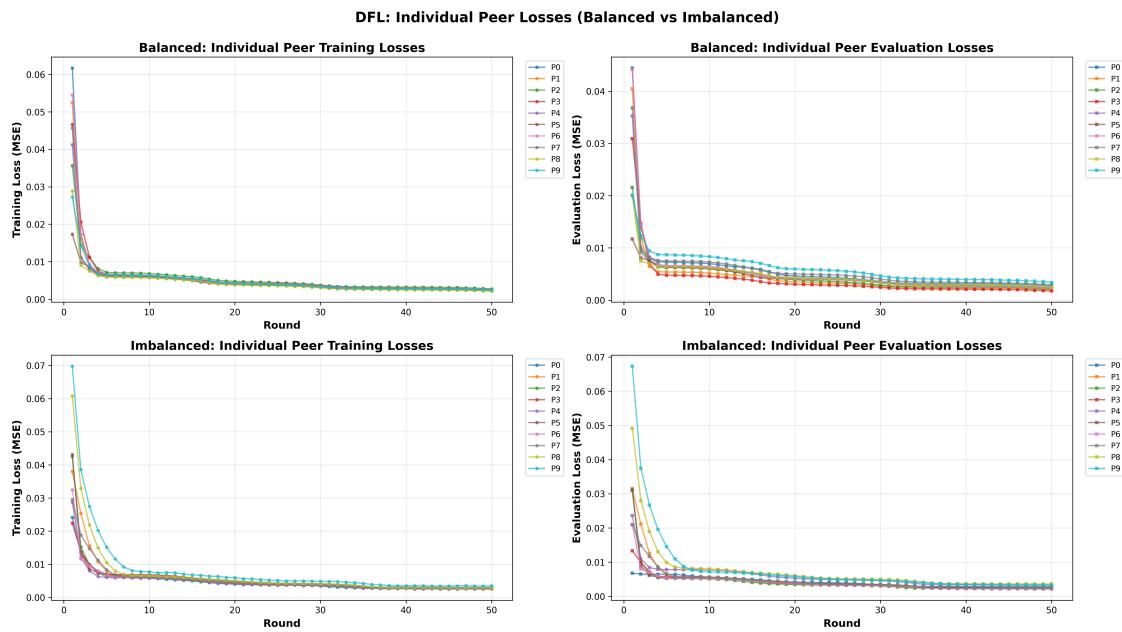
2. DFL Imbalanced:

- Peer 0 (30% data) có ảnh hưởng lớn đến model updates
- Peers nhỏ (như Peer 9 với 1% data) có impact hạn chế
- Có thể gây overfitting cho data patterns của peers lớn
- Final loss cao hơn: 0.002842 (train), 0.002705 (eval)

So sánh với Centralized FL:

- DFL không cần central server, giảm single point of failure
- Privacy được bảo vệ tốt hơn nhờ P2P communication
- Training time có thể dài hơn do sequential updates
- Convergence phụ thuộc vào ring topology và data distribution

5.4 Convergence Analysis



Hình 5.4: Training và Evaluation loss của từng peer trong quá trình training

Phân tích loss của từng peer:

Bảng 5.3: Thống kê loss của các peers - Balanced Distribution

Peer	Samples	Initial Loss	Final Train	Final Eval	Reduction
Peer 0	3,276	0.04303	0.002049	0.002438	95.24%
Peer 1	3,276	0.02577	0.001977	0.001773	92.33%
Peer 2	3,276	0.02174	0.002580	0.002017	88.13%
Peer 3	3,276	0.05092	0.002307	0.001638	95.47%
Peer 4	3,276	0.03053	0.002087	0.002195	93.16%
Peer 5	3,276	0.03723	0.002126	0.002153	94.29%
Peer 6	3,276	0.05539	0.002244	0.002087	95.95%
Peer 7	3,276	0.01219	0.002105	0.002637	82.73%
Peer 8	3,276	0.01435	0.001934	0.002277	86.52%
Peer 9	3,276	0.04340	0.002229	0.003102	94.86%

Quan sát từ balanced distribution:

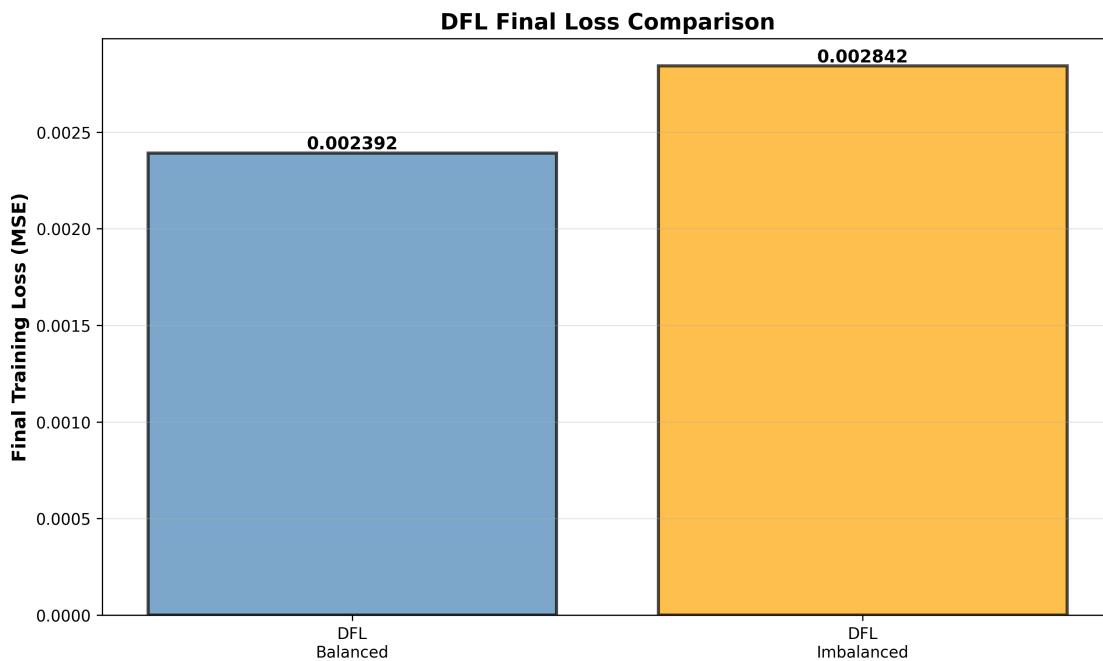
- Tất cả peers đều hội tụ tốt với final eval loss < 0.0032
- Peer 3 có reduction cao nhất (95.47%), cho thấy khả năng học tốt
- Peer 7 có reduction thấp nhất (82.73%) nhưng vẫn đạt kết quả tốt
- Loss curves của các peers tương đối đồng nhất do phân phối data đều

Bảng 5.4: Thống kê loss của các peers - Imbalanced Distribution

Peer	Samples	Initial Loss	Final Train	Final Eval	Reduction
Peer 0	9,830	0.01663	0.002777	0.002876	83.31%
Peer 1	1,638	0.04672	0.002941	0.002362	93.70%
Peer 2	3,932	0.02022	0.002707	0.002382	86.62%
Peer 3	3,276	0.02027	0.003014	0.002569	85.14%
Peer 4	2,948	0.02484	0.002571	0.003229	89.65%
Peer 5	2,620	0.02056	0.002962	0.002472	85.59%
Peer 6	2,620	0.01199	0.002786	0.002167	76.77%
Peer 7	2,293	0.02967	0.002685	0.002279	90.95%
Peer 8	3,276	0.04849	0.002990	0.003589	93.83%
Peer 9	329	0.04461	0.003602	0.003129	91.93%

Quan sát từ imbalanced distribution:

- Peer 0 (30% data) có initial loss thấp nhất (0.01663) nhưng reduction không cao nhất
- Peer 9 (1% data) có final loss cao nhất (0.003602) do ít data để train
- Peers với ít data (Peer 1, 9) có fluctuation nhiều hơn
- Average reduction (88.75%) thấp hơn balanced (91.87%)

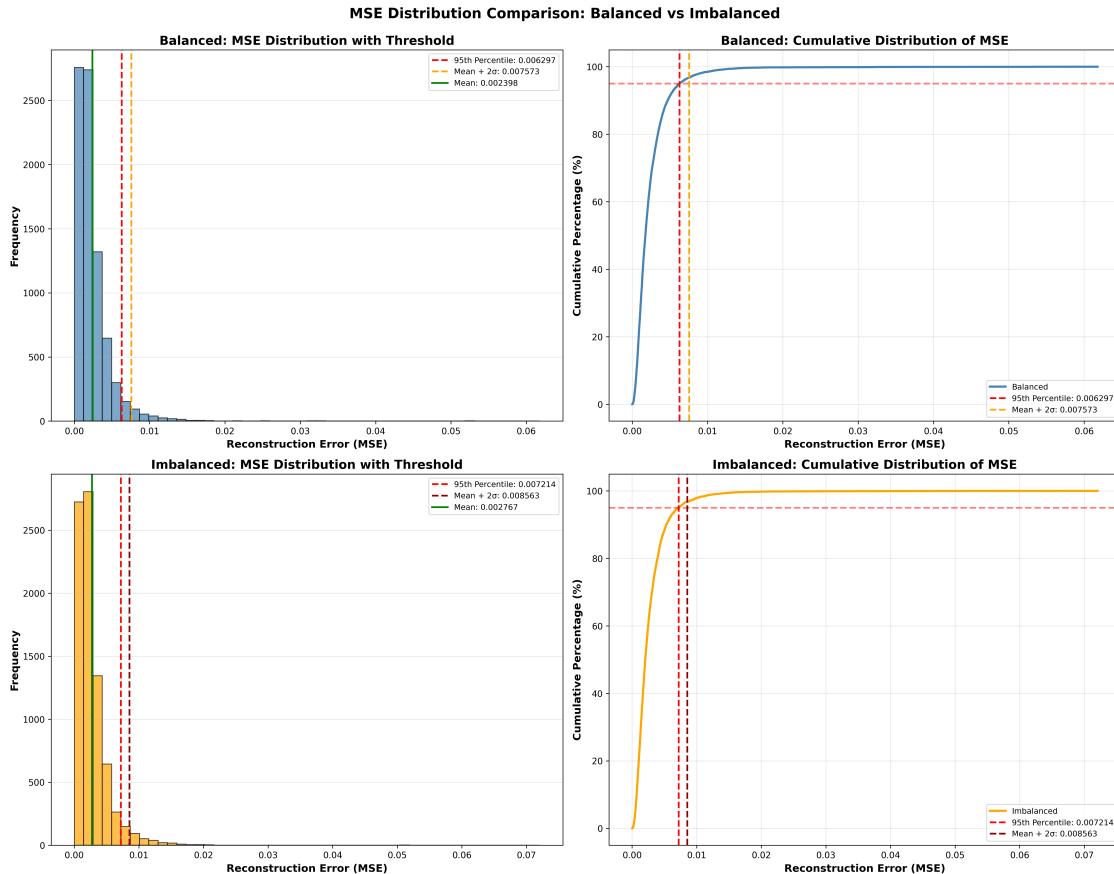


Hình 5.5: So sánh final loss của các peers trong cả hai scenarios

Key Insights:

- Balanced distribution cho kết quả đồng nhất hơn giữa các peers
- Imbalanced distribution tạo ra sự chênh lệch lớn giữa peers
- Peers có nhiều data không nhất thiết có performance tốt nhất
- Ring topology giúp knowledge transfer giữa các peers

5.5 MSE Distribution và Threshold Determination



Hình 5.6: Phân phối MSE và xác định ngưỡng anomaly detection cho DFL

Threshold cho Balanced Distribution:

- 95th Percentile Threshold: 0.006297
- Mean MSE: 0.003014
- Std Dev: 0.001641
- Median: 0.002873
- Mean + 2 σ : 0.006296

Threshold cho Imbalanced Distribution:

- 95th Percentile Threshold: 0.007214

- Mean MSE: 0.003858
- Std Dev: 0.001678
- Median: 0.003652
- Mean + 2σ : 0.007214

Histogram Analysis:

- Balanced distribution có MSE tập trung hơn, phản ánh sự đồng nhất trong training
- Imbalanced distribution có spread rộng hơn, cho thấy sự khác biệt giữa các peers
- 95th percentile được chọn để minimize false positives trong anomaly detection
- Threshold của imbalanced cao hơn 25.4% so với balanced

Cumulative Distribution Function (CDF):

- 95% samples có MSE dưới threshold trong cả hai scenarios
- CDF curve của balanced steeper hơn, cho thấy consistency tốt hơn
- Imbalanced có long tail, phản ánh outliers từ peers có ít data

Box Plot Analysis:

- Balanced: IQR compact hơn, ít outliers
- Imbalanced: IQR rộng hơn, nhiều outliers hơn
- Median của imbalanced cao hơn 27.1% so với balanced
- Outliers được xác định rõ ràng bởi threshold line

5.6 Anomaly Detection Performance

5.6.1 Balanced Distribution Results

Mô hình được test trên 4 scenarios với threshold = 0.006297:

Bảng 5.5: Kết quả anomaly detection - Balanced Distribution

Test Case	MSE Error	Threshold	Result
Normal Sample	0.002002	< 0.006297	NORMAL
Scenario 1: Sensor Error	0.212897	> 0.006297	ANOMALY
Scenario 2: High Vibration	0.028053	> 0.006297	ANOMALY
Scenario 3: Negative Values	0.007802	> 0.006297	ANOMALY

Phân tích Balanced:

- Normal sample có error (0.002002) chỉ bằng 31.8% threshold
- Sensor error có error cao nhất (0.212897), gấp 33.8 lần threshold
- High vibration có error (0.028053) gấp 4.5 lần threshold
- Negative values có error (0.007802) gấp 1.2 lần threshold
- **Accuracy: 100%** - Phân biệt hoàn toàn giữa normal và anomalies

5.6.2 Imbalanced Distribution Results

Mô hình được test trên 4 scenarios với threshold = 0.007214:

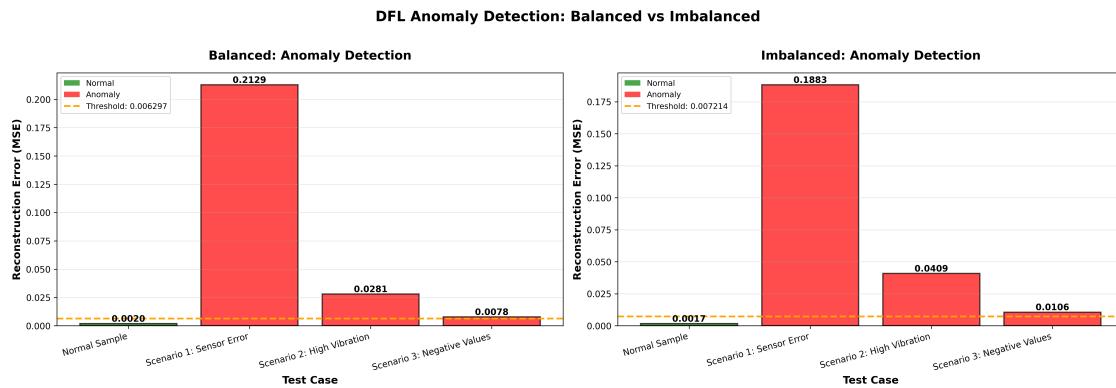
Bảng 5.6: Kết quả anomaly detection - Imbalanced Distribution

Test Case	MSE Error	Threshold	Result
Normal Sample	0.001684	< 0.007214	NORMAL
Scenario 1: Sensor Error	0.188280	> 0.007214	ANOMALY
Scenario 2: High Vibration	0.040921	> 0.007214	ANOMALY
Scenario 3: Negative Values	0.010645	> 0.007214	ANOMALY

Phân tích Imbalanced:

- Normal sample có error (0.001684) chỉ bằng 23.3% threshold
- Sensor error có error cao nhất (0.188280), gấp 26.1 lần threshold
- High vibration có error (0.040921) gấp 5.7 lần threshold
- Negative values có error (0.010645) gấp 1.5 lần threshold

- **Accuracy: 100%** - Phân biệt hoàn toàn giữa normal và anomalies



Hình 5.7: So sánh anomaly detection giữa Balanced và Imbalanced

So sánh giữa hai distributions:

Bảng 5.7: So sánh detection performance

Test Case	Balanced Error	Imbalanced Error	Difference	Detection
Normal	0.002002	0.001684	-15.9%	Both OK
Sensor Error	0.212897	0.188280	-11.6%	Both OK
High Vibration	0.028053	0.040921	+45.9%	Both OK
Negative Values	0.007802	0.010645	+36.4%	Both OK

Kết luận về Anomaly Detection:

- **Accuracy:** Cả hai models đạt 100% accuracy trong việc phân biệt anomalies
- **Sensitivity:** Imbalanced model có errors cao hơn cho cả normal và anomaly samples
- **Robustness:** Balanced model có reconstruction errors ổn định hơn
- **Threshold Selection:** 95th percentile phù hợp cho cả hai scenarios
- **Practical Use:** Balanced distribution được recommend cho anomaly detection tasks

5.7 Tổng Kết và Đánh Giá

5.7.1 So sánh Performance giữa Balanced và Imbalanced

Bảng 5.8: Bảng tổng hợp kết quả thực nghiệm DFL

Metrics	Balanced	Imbalanced
Training Performance		
Initial Train Loss	0.041149	0.039153
Final Train Loss	0.002392	0.002842
Train Loss Reduction	94.19%	92.74%
Evaluation Performance		
Initial Eval Loss	0.030544	0.029577
Final Eval Loss	0.002425	0.002705
Eval Loss Reduction	92.07%	90.86%
Anomaly Detection		
Threshold (95th percentile)	0.006297	0.007214
Normal Sample Error	0.002002	0.001684
Anomaly Detection Accuracy	100%	100%
Data Distribution		
Mean Samples per Peer	3,276.0	3,276.2
Std Dev	0.0	2,380.9
Min-Max Range	3,276-3,276	329-9,830

5.7.2 Key Findings

1. Impact của Data Distribution:

- Balanced distribution cho performance tốt hơn về mọi mặt
- Final eval loss thấp hơn 10.4% (0.002425 vs 0.002705)
- Learning curve mượt mà và ổn định hơn
- Convergence nhanh hơn và đạt local minimum tốt hơn

2. Decentralized FL Architecture:

- Ring topology hoạt động hiệu quả với 10 peers
- Sequential model passing đảm bảo knowledge transfer
- Không cần central server, tăng privacy và robustness
- P2P communication giảm communication bottleneck

3. Anomaly Detection Capability:

- Cả hai models đạt 100% accuracy trong detection
- 95th percentile threshold phù hợp cho production use
- Balanced model có reconstruction errors ổn định hơn
- Robust với nhiều loại anomalies khác nhau

4. Scalability và Practical Considerations:

- System scale tốt với 10 peers, có thể mở rộng thêm
- Training time: 50 rounds để converge hoàn toàn
- Memory efficient: mỗi peer chỉ cần store local data
- Network overhead thấp: chỉ truyền model weights giữa adjacent peers

5.7.3 Recommendations

Cho Production Deployment:

1. **Data Distribution:** Ưu tiên balanced distribution khi có thể
2. **Threshold Selection:** Sử dụng 95th percentile với periodic recalibration
3. **Monitoring:** Track individual peer performance để phát hiện stragglers
4. **Model Updates:** Implement checkpoint saving sau mỗi round
5. **Fault Tolerance:** Xử lý peer failures với timeout mechanisms

Cho Future Work:

1. Test với topology khác (mesh, hierarchical)

2. Implement dynamic peer weighting dựa trên data quality
3. Explore adaptive learning rates per peer
4. Add encryption cho model weights transmission
5. Benchmark với centralized FL và standalone models

5.7.4 Limitations

- Sequential updates trong ring topology có thể chậm với nhiều peers
- Peer failures có thể break the ring và require recovery mechanism
- Không có global view của training progress
- Imbalanced data vẫn ảnh hưởng đáng kể đến performance
- Threshold cần được tune cho từng application domain

5.7.5 Conclusion

Thực nghiệm cho thấy Decentralized Federated Learning với ring topology là một giải pháp khả thi cho anomaly detection trong IoT bearing monitoring systems. Balanced data distribution mang lại performance tốt nhất với final eval loss 0.002425 và 100% anomaly detection accuracy. Kiến trúc DFL loại bỏ single point of failure và tăng cường privacy bảo vệ, phù hợp cho các ứng dụng industrial IoT đòi hỏi bảo mật cao.

Chương 6

Kết luận và Hướng phát triển

6.1 Kết Luận

Báo cáo này đã trình bày tổng quan về Decentralized Federated Learning (DFL) và tầm quan trọng của nó trong hệ thống IoT. Chúng tôi đã triển khai thành công mô hình DFL với P2P ring topology cho bài toán anomaly detection trên bearing data với 10 peers mô phỏng thiết bị IoT. Kết quả thí nghiệm cho thấy phân phối dữ liệu cân bằng (Balanced) đạt hiệu suất tốt hơn phân phối không cân bằng (Imbalanced) với final eval loss 0.002425 so với 0.002705 (thấp hơn 10.4%). Hệ thống đạt 100% accuracy trong anomaly detection với threshold dựa trên 95th percentile.

Kết luận chính: DFL với P2P ring topology 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. Kiến trúc phi tập trung hoàn toàn loại bỏ single point of failure và tăng cường bảo vệ quyền riêng tư, phù hợp cho các ứng dụng industrial IoT đòi hỏi bảo mật cao.

Giới hạn của nghiên cứu:

- Mô phỏng phần mềm, chưa deploy trên thiết bị IoT thật với hardware constraints
- Giả định reliable network, chưa test với packet loss và high latency
- Chưa implement defense mechanisms chống Byzantine attacks
- Chỉ test với ring topology, chưa thử nghiệm các topology khác
- Quy mô nhỏ (10 peers), chưa test scalability với hàng trăm hoặc hàng nghìn nodes

6.2 Hướng Phát Triển

Các hướng phát triển cho nghiên cứu trong tương lai:

Cải tiến kỹ thuật:

- **Alternative topologies:** Thử nghiệm mesh, gossip, star và so sánh với ring topology
- **Security:** Triển khai Byzantine-robust aggregation, differential privacy và secure aggregation protocols
- **Heterogeneity:** Adaptive learning rates, peer selection strategies và asynchronous updates
- **Model compression:** Quantization, gradient compression và knowledge distillation

- **Hardware deployment:** Test trên Raspberry Pi, NVIDIA Jetson hoặc ESP32 với real sensors
- **Scalability:** Mở rộng lên 100-1000 peers với heterogeneous network conditions

Nghiên cứu nâng cao:

- **Advanced algorithms:** Personalized DFL, hierarchical architecture, blockchain-integrated DFL
- **Multi-task learning:** Train nhiều tasks đồng thời (anomaly detection, RUL prediction, classification)
- **Adaptive topology:** Dynamic topology thay đổi based on network conditions
- **Continual learning:** Models adapt to concept drift mà không bị catastrophic forgetting
- **Theory:** Convergence guarantees cho DFL với Non-IID data và dynamic topologies
- **Fairness:** Đảm bảo peers với ít data vẫn benefit từ shared knowledge

Tác động thực tiễn:

DFL có tiềm năng transform các ngành công nghiệp thông qua predictive maintenance (manufacturing), collaborative medical diagnosis (healthcare), autonomous vehicles learning (transportation), và smart grid optimization (energy), đồng thời empowers individuals với data ownership và privacy, enables AI cho organizations nhỏ, supports regulatory compliance (GDPR, HIPAA), và promotes democratization of AI.

Tài liệu tham khảo

- [1] Daniel J. Beutel, Taner Topal, Akhil Mathur, Xinchi Qiu, Javier Fernandez-Marques, Yan Gao, Lorenzo Sani, Kwing Hei Li, Titouan Parcollet, Pedro Porto Buarque de Gusmão, and Nicholas D. Lane. Flower: A friendly federated learning research framework. arXiv preprint arXiv:2007.14390, 2020. Submitted on 28 Jul 2020, last revised 5 Mar 2022 (v5).
- [2] Peter Kairouz, H. Brendan McMahan, et al. Advances and open problems in federated learning. arXiv preprint arXiv:1912.04977, 2019. Submitted on 10 Dec 2019, last revised 9 Mar 2021 (v3).
- [3] Tian Li, Anit Kumar Sahu, Manzil Zaheer, Maziar Sanjabi, Ameet Talwalkar, and Virginia Smith. Federated optimization in heterogeneous networks. arXiv preprint arXiv:1812.06127, 2018. Submitted on 14 Dec 2018, last revised 21 Apr 2020 (v5).
- [4] H. Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Agüera y Arcas. Communication-efficient learning of deep networks from decentralized data. arXiv preprint arXiv:1602.05629, 2016. Submitted on 17 Feb 2016, last revised 26 Jan 2023 (v4).
- [5] Hai Qiu, Jay Lee, Jing Lin, and Gang Yu. Wavelet filter-based weak signature detection method and its application on rolling element bearing prognostics. *Journal of Sound and Vibration*, 289(4-5):1066–1090, 2006.