# Báo Cáo Đồ Án

Federated Learning cho Phát Hiện Bất Thường
Trong Dữ Liệu Vòng Bi Sử Dụng Autoencoder

Sinh viên thực hiện

Chương trình: Thạc sĩ Khoa học Dữ liệu

Ngày 21 tháng 11 năm 2025

**Tóm tắt nội dung**

Báo cáo này trình bày kết quả triển khai hệ thống Federated Learning (FL) để phát hiện bất thường trong dữ liệu cảm biến vòng bi công nghiệp sử dụng mô hình Autoencoder. Nghiên cứu thực hiện hai thí nghiệm với phân phối dữ liệu cân bằng (IID) và không cân bằng (Non-IID) để đánh giá hiệu suất của thuật toán FedAvg trong điều kiện thực tế. Kết quả cho thấy mô hình đạt được hiệu suất tốt với cả hai phân phối dữ liệu, trong đó phân phối không cân bằng cho kết quả tốt hơn với loss cuối cùng là 0.002755 so với 0.003859 của phân phối cân bằng.

## 1 Giới Thiệu

### 1.1 Bối Cảnh

Trong kỷ nguyên Công nghiệp 4.0, bảo trì dự đoán (predictive maintenance) đóng vai trò quan trọng trong việc giảm thiểu thời gian ngừng hoạt động và chi phí bảo trì thiết bị. Vòng bi là một trong những thành phần quan trọng nhất trong máy móc công nghiệp, và việc phát hiện sớm các bất thường có thể ngăn ngừa hỏng hóc nghiêm trọng.

### 1.2 Thách Thức

Các thách thức chính trong triển khai hệ thống phát hiện bất thường phân tán:

- **Bảo mật dữ liệu**: Dữ liệu cảm biến không thể chia sẻ trực tiếp do quy định bảo mật
- **Phân phối dữ liệu không đều**: Các thiết bị khác nhau thu thập lượng dữ liệu khác nhau
- **Giới hạn tài nguyên**: Các thiết bị edge có tài nguyên tính toán hạn chế

### 1.3 Giải Pháp Đề Xuất

Sử dụng Federated Learning với framework Flower để:

- Huấn luyện mô hình phân tán mà không chia sẻ dữ liệu thô
- So sánh hiệu suất với dữ liệu cân bằng và không cân bằng
- Triển khai mô hình Autoencoder để học các pattern bình thường

# 2 Phương Pháp

## 2.1 Kiến Trúc Hệ Thống

### 2.1.1 Mô Hình Autoencoder

Kiến trúc mô hình được thiết kế để nén và tái tạo dữ liệu cảm biến 8 chiều:

- **Encoder**: $8 \rightarrow 4 \rightarrow 2$ (nén dữ liệu)
- **Decoder**: $2 \rightarrow 4 \rightarrow 8$ (tái tạo dữ liệu)
- **Hàm kích hoạt**: ReLU cho các lớp ẩn
- **Hàm loss**: Mean Squared Error (MSE)

Listing 1: Định nghĩa kiến trúc Autoencoder

```
class BearingAutoencoder(nn.Module):
    def __init__(self):
        super().__init__()
        # Encoder: 8 -> 4 -> 2
        self.encoder = nn.Sequential(
            nn.Linear(8, 4),
            nn.ReLU(),
            nn.Linear(4, 2),
            nn.ReLU()
        )
        # Decoder: 2 -> 4 -> 8
        self.decoder = nn.Sequential(
            nn.Linear(2, 4),
            nn.ReLU(),
            nn.Linear(4, 8)
        )
```

### 2.1.2 Federated Learning với Flower

Hệ thống FL được cấu hình với các thông số:

Bảng 1: Cấu hình Federated Learning

| **Tham số** | **Giá trị** |
|---|---|
| Số lượng clients | 10 |
| Số vòng FL | 20 |
| Local epochs | 1 |
| Learning rate | 0.001 |
| Optimizer | Adam |
| Batch size | 32 |
| Chiến lược tổng hợp | FedAvg |

## 2.2 Dữ Liệu

### 2.2.1 Nguồn Dữ Liệu

Dữ liệu cảm biến vòng bi từ NASA IMS Bearing Dataset bao gồm:

- 8 kênh đo rung động (4 vòng bi × 2 cảm biến/vòng bi)
- Tổng số mẫu: 32,768 samples
- Phân chia: 80% training, 20% testing

#### 2.2.2 Phân Phối Dữ Liệu

Hai kịch bản thí nghiệm:

Bảng 2: Phân phối dữ liệu giữa các clients

| Client ID | Exp 1: Cân bằng | Exp 2: Không cân bằng |
|---|---|---|
| Client 0 | 10% | 20% |
| Client 1 | 10% | 15% |
| Client 2 | 10% | 12% |
| Client 3 | 10% | 10% |
| Client 4 | 10% | 9% |
| Client 5 | 10% | 8% |
| Client 6 | 10% | 8% |
| Client 7 | 10% | 7% |
| Client 8 | 10% | 6% |
| Client 9 | 10% | 5% |

### 2.3 Quy Trình Huấn Luyện

#### 2.3.1 Thuật Toán FedAvg

Federated Averaging tổng hợp các tham số mô hình theo công thức:

$$\theta_{global} = \sum_{i=1}^{N} \frac{n_i}{N_{total}} \cdot \theta_i \tag{1}$$

Trong đó:

- $\theta_{global}$: Tham số mô hình toàn cục
- $\theta_i$: Tham số từ client $i$
- $n_i$: Số mẫu training của client $i$
- $N_{total}$: Tổng số mẫu training

#### 2.3.2 Chu Trình Huấn Luyện

1. Server khởi tạo mô hình toàn cục
2. Phân phối mô hình cho các clients
3. Mỗi client huấn luyện local với dữ liệu riêng
4. Clients gửi cập nhật về server
5. Server tổng hợp theo FedAvg
6. Lặp lại cho 20 rounds

# 3 Kết Quả Thực Nghiệm

## 3.1 Thí Nghiệm 1: Dữ Liệu Cân Bằng (IID)

### 3.1.1 Cấu Hình

- Mỗi client có 3,276 samples training và 820 samples testing
- Phân phối đều: 10% dữ liệu cho mỗi client
- Điều kiện lý tưởng để đánh giá baseline

### 3.1.2 Kết Quả

Bảng 3: Kết quả Thí nghiệm 1 (Dữ liệu cân bằng)

| **Metric** | **Giá trị đầu** | **Giá trị cuối** |
|---|---|---|
| Training Loss | 0.0231 | 0.003859 |
| Evaluation Loss | 0.0179 | 0.003954 |
| Độ giảm Loss | - | 83.3% |

## 3.2 Thí Nghiệm 2: Dữ Liệu Không Cân Bằng (Non-IID)

### 3.2.1 Cấu Hình

- Client 0: 6,553 samples (20%)
- Client 9: 1,638 samples (5%)
- Mô phỏng kịch bản thực tế với phân phối không đều

### 3.2.2 Kết Quả

Bảng 4: Kết quả Thí nghiệm 2 (Dữ liệu không cân bằng)

| **Metric** | **Giá trị đầu** | **Giá trị cuối** |
|---|---|---|
| Training Loss | 0.0359 | 0.002755 |
| Evaluation Loss | 0.0175 | 0.002687 |
| Độ giảm Loss | - | 92.3% |

## 3.3 So Sánh Hai Thí Nghiệm

### 3.3.1 Phân Tích Định Lượng

Bảng 5: So sánh hiệu suất giữa hai thí nghiệm

| **Thí nghiệm** | **Train Loss** | **Eval Loss** | **Chênh lệch** |
|---|---|---|---|
| Exp 1: Cân bằng | 0.003859 | 0.003954 | Baseline |
| Exp 2: Không cân bằng | 0.002755 | 0.002687 | -28.59% |

### 3.3.2 Quan Sát Chính

- **Kết quả bất ngờ**: Dữ liệu không cân bằng cho hiệu suất tốt hơn 28.59%

- **Hội tụ**: Cả hai thí nghiệm đều hội tụ ổn định sau 20 rounds
- **Overfitting**: Không có dấu hiệu overfitting nghiêm trọng
- **Generalization**: Evaluation loss gần với training loss

### 3.4 Phân Tích Kết Quả

#### 3.4.1 Tại Sao Dữ Liệu Không Cân Bằng Lại Tốt Hơn?

Có thể giải thích bởi các yếu tố sau:

1. **Đa dạng dữ liệu**: Clients lớn (20%) có nhiều patterns, clients nhỏ (5%) giúp regularization
2. **Weighted averaging**: FedAvg cho trọng số lớn hơn với clients có nhiều dữ liệu
3. **Natural regularization**: Sự không đồng nhất giúp tránh overfitting
4. **Dataset characteristics**: Đặc điểm riêng của dữ liệu vòng bi

#### 3.4.2 Đánh Giá Độ Tin Cậy

- **Stability**: Loss giảm đều đặn qua các rounds
- **Convergence**: Cả hai experiments hội tụ tốt
- **Generalization gap**: Nhỏ ($< 0.001$), cho thấy mô hình generalize tốt

## 4 Kết Luận và Hướng Phát Triển

### 4.1 Kết Luận

1. **Thành công**: Triển khai thành công hệ thống FL cho phát hiện bất thường vòng bi
2. **Hiệu suất**: Đạt training loss $< 0.003$ và evaluation loss $< 0.004$
3. **Non-IID advantage**: Dữ liệu không cân bằng cho kết quả tốt hơn dự kiến
4. **Khả năng triển khai**: Hệ thống sẵn sàng cho môi trường production

### 4.2 Đóng Góp

- So sánh chi tiết giữa phân phối IID và Non-IID
- Triển khai hoàn chỉnh FL cho dữ liệu cảm biến công nghiệp
- Framework có thể mở rộng cho các loại cảm biến khác

### 4.3 Hạn Chế

- Chưa test với dữ liệu anomaly thực tế
- Số lượng rounds và clients còn hạn chế
- Chưa thử nghiệm các chiến lược FL khác (FedProx, FedAdam)

### 4.4 Hướng Phát Triển

1. **Mở rộng experiments**:
   - Test với nhiều rounds hơn (50-100 rounds)
   - Thử nghiệm với số lượng clients khác nhau (20, 50, 100)
   - So sánh với các thuật toán FL khác: FedProx, FedAdam, SCAFFOLD
2. **Cải thiện mô hình**:
   - Thử nghiệm với kiến trúc Autoencoder sâu hơn
   - Sử dụng Variational Autoencoder (VAE)
   - Tích hợp attention mechanism
3. **Triển khai thực tế**:
   - Test với dữ liệu anomaly thực tế
   - Triển khai trên edge devices
   - Xây dựng dashboard monitoring
   - Tích hợp hệ thống cảnh báo real-time
4. **Bảo mật và Privacy**:
   - Tích hợp Differential Privacy
   - Secure aggregation
   - Homomorphic encryption

### 4.5 Ứng Dụng Thực Tiễn

- **Công nghiệp 4.0**: Bảo trì dự đoán cho nhà máy thông minh
- **IoT phân tán**: Giám sát thiết bị từ xa
- **Healthcare**: Giám sát thiết bị y tế
- **Transportation**: Giám sát tình trạng phương tiện

## Tài Liệu Tham Khảo


1. McMahan, H. B., et al. (2017). "Communication-efficient learning of deep networks from decentralized data." *AISTATS*.
2. Beutel, D. J., et al. (2020). "Flower: A friendly federated learning framework." *arXiv preprint arXiv:2007.14390*.
3. Li, T., et al. (2020). "Federated optimization in heterogeneous networks." *MLSys*.
4. Qiu, H., et al. (2006). "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.
5. Kairouz, P., et al. (2021). "Advances and open problems in federated learning." *Foundations and Trends in Machine Learning*, 14(1-2), 1-210.