Federated Learning trong gian lận

OK. Dưới đây là **BẢN TÓM TẮT CHÍNH THỨC ĐỂ TRUYỀN ĐẠT CHO CẢ NHÓM**
(đọc là hiểu, thống nhất tư duy, tránh hiểu sai khi làm & khi bảo vệ)

FEDERATED LEARNING TRONG PHÁT HIỆN GIAN LẬN NGÂN HÀNG

Tóm tắt ý tưởng – workflow – giải đáp thắc mắc

---

1. Mục tiêu đề tài

- Phát hiện **gian lận giao dịch tài chính**
- Dữ liệu ngân hàng **không thể chia sẻ**
- Nhiều ngân hàng vẫn muốn **học chung pattern gian lận**

➡ **Federated Learning (FL)** cho phép huấn luyện chung **mà không gom dữ liệu**

---

2. Mô hình được chọn (QUYẾT ĐỊNH CHUNG)

👉 Neural Network (hoặc Logistic Regression)

Lý do:

- Gradient-based → dùng **FedAvg chuẩn**
- Dễ triển khai, dễ giải thích
- Phù hợp fraud detection

🚫 Không chọn Random Forest cho workflow chính
(vì không tổng hợp tham số được)

---

3. Dữ liệu & nhãn gian lận

3.1 Dữ liệu ở mỗi client (ngân hàng)

- Giao dịch nội bộ
- Không chia sẻ ra ngoài

3.2 Nhãn gian lận từ đâu?

Không phải tự gán thủ công cho đề tài, mà mô phỏng theo thực tế:

- Khiếu nại khách hàng
- Chargeback
- Nhân viên xác minh
- Rule-based system

➡ Sau một thời gian:

- `1` = fraud
- `0` = normal

📌 **FL không tạo nhãn – chỉ dùng nhãn có sẵn**

---

4. VẤN ĐỀ QUAN TRỌNG: DỮ LIỆU KHÁC NHAU THÌ SAO?

❌ Không được:

- Mỗi client có **feature vector khác nhau**
- Khác số cột / khác ý nghĩa cột

✅ Được (CÁCH ĐÚNG):

- **Raw data có thể khác**
- Nhưng **SAU FEATURE ENGINEERING → vector đặc trưng PHẢI GIỐNG NHAU**

Ví dụ schema chung:

```
powershell

amount
transaction_hour
merchant_category
device_type
geo_distance
tx_count_last_24h
```

- Thiếu feature → điền 0 / unknown

➡ Đây là cách **thực tế người ta làm**

---

5. WORKFLOW CHI TIẾT (THỐNG NHẤT)

Bước 1 – Khởi tạo

- Server tạo mô hình Neural Network ban đầu

Bước 2 – Phân phối

- Server gửi **cùng 1 model** cho tất cả client

Bước 3 – Huấn luyện cục bộ (Client)

Tại mỗi ngân hàng:

- Dữ liệu nội bộ + nhãn
- Feature đã chuẩn hóa
- Huấn luyện NN bằng SGD/Adam

Kết quả:

```arduino
Client i → weight w_i
```

⛔ Không gửi dữ liệu

Bước 4 – Gửi cập nhật

- Client gửi **tham số model / gradient**
- Có thể mã hóa

Bước 5 – Tổng hợp (Server)

Server dùng **FedAvg**:

```ini
w_global = Σ (n_i / N) * w_i
```

Bước 6 – Lặp nhiều vòng

- Gửi lại model
- Train tiếp
- Đến khi hội tụ

Bước 7 – Triển khai

- Model cuối chạy tại mỗi ngân hàng
- Dự đoán gian lận real-time

---

6. CÁC THẮC MẮC & CÂU TRẢ LỜI CHUẨN

❓ Client & server có huấn luyện khác nhau không?

➡ Không

- Cùng mô hình
- Cùng loss
- Cùng optimizer

❓ Dataset khác nhau có train chung được không?

➡ Chỉ khi feature vector giống nhau

- Raw khác: OK
- Feature khác: KHÔNG

❓ Có dùng Random Forest được không?

➡ Có thể nhưng:

- Không phải FL chuẩn
- Phức tạp
- Không chọn cho đề tài chính

❓ Thực tế ngân hàng có làm vậy không?

➡ Có

- Dùng cùng kiến trúc model

- Khác nhau chỉ ở dữ liệu

❓ FL có an toàn tuyệt đối không?

➡ Không 100%

- Nhưng an toàn hơn centralized
- Có thể kết hợp secure aggregation

---

7. CÂU CHỐT Ý TƯỞNG (THỐNG NHẤT TOÀN NHÓM)

> Đề tài sử dụng Federated Learning với mô hình Neural Network để phát hiện gian lận giao dịch tài chính, trong đó mỗi ngân hàng huấn luyện mô hình cục bộ trên dữ liệu đã gán nhãn nội bộ và chỉ chia sẻ tham số mô hình, đảm bảo quyền riêng tư và bảo mật dữ liệu.

---

8. KẾT LUẬN CHO NHÓM

- ✔️ Ý tưởng đúng thực tế
- ✔️ Workflow rõ ràng
- ✔️ Không mâu thuẫn lý thuyết
- ✔️ Dễ bảo vệ
