



BIGDATA

Sinh viên:

Phan Văn Tài 2202081

Phan Minh Thuy 2202079

Dự án cuối cùng Xây dựng hệ thống phát hiện gian lận thẻ tín dụng real-time

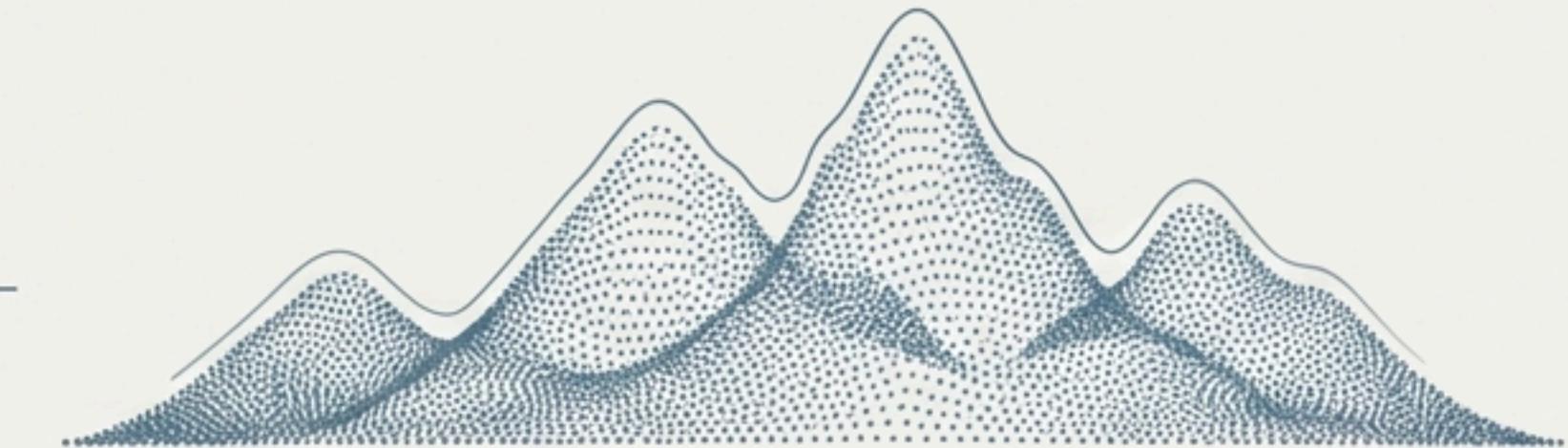
Giảng viên hướng dẫn: Ts. Cao Tiến Dũng



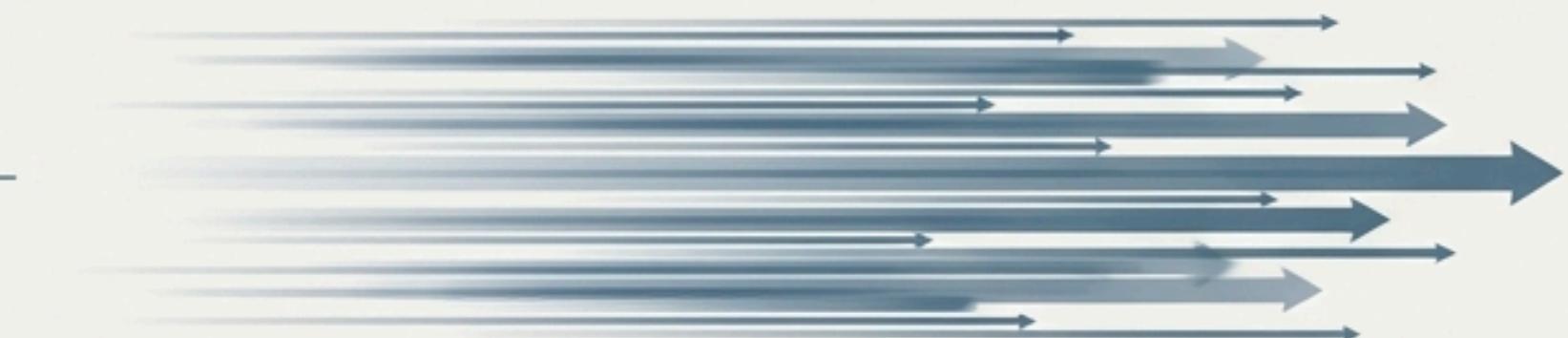


Thách thức của bài toán phát hiện gian lận

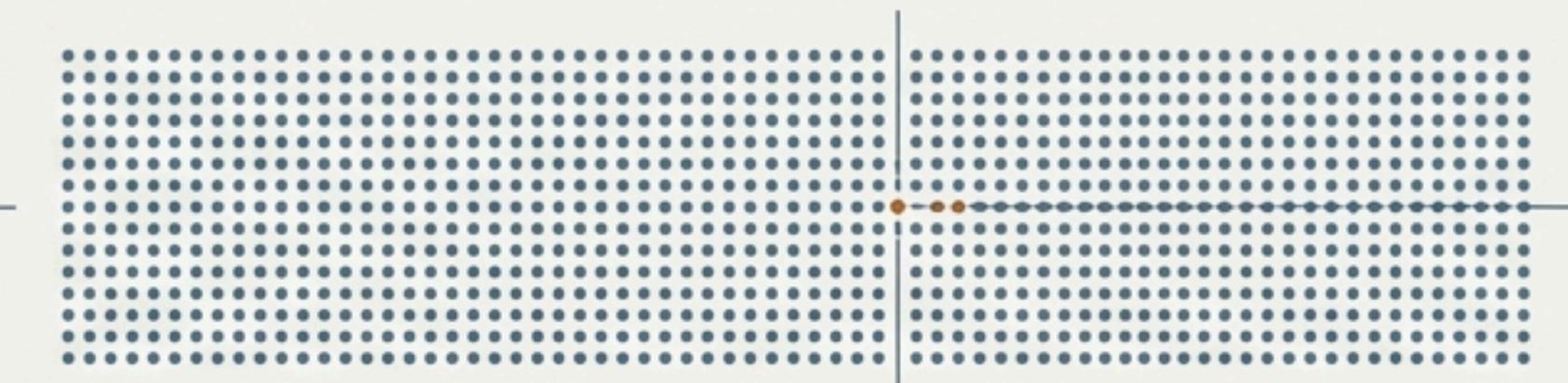
- **Bản chất vấn đề:** Các giao dịch bất hợp pháp, không được chủ thẻ cho phép, cần được ngăn chặn ngay khi chúng xảy ra.
- **Yêu cầu cốt lõi:** Việc phát hiện phải diễn ra gần như tức thời (near real-time) để có thể khóa thẻ hoặc hủy giao dịch kịp thời.



Dữ liệu cực lớn (Volume): Hệ thống phải xử lý hàng triệu giao dịch mỗi ngày.



Dữ liệu dạng luồng (Velocity): Giao dịch đến liên tục, đòi hỏi xử lý ngay khi phát sinh thay vì chờ đợi.

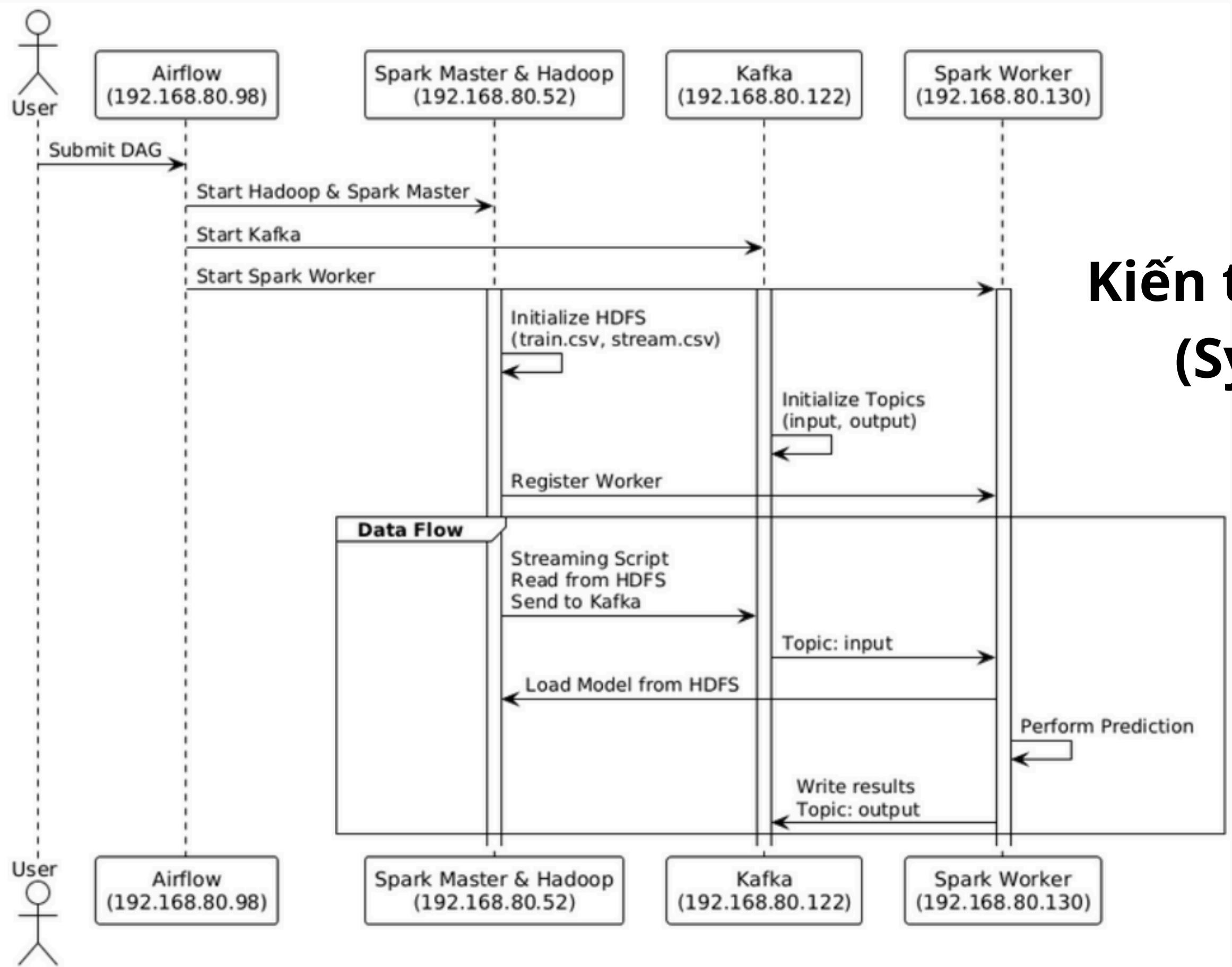


Dữ liệu mất cân bằng (Imbalance): Tỉ lệ giao dịch gian lận (< 0.2%) là cực kỳ nhỏ so với giao dịch hợp lệ, gây khó khăn cho việc huấn luyện mô hình chính xác.



Kiến trúc hệ thống và các công nghệ cốt lõi

-  **Apache Hadoop (HDFS):** Hệ thống file phân tán để lưu trữ dữ liệu huấn luyện lớn và mô hình Machine Learning đã huấn luyện.
-  **Apache Spark:** Nền tảng xử lý dữ liệu phân tán, đóng 2 vai trò chính:
 - **Spark ML:** Huấn luyện mô hình phân loại (Random Forest) trên dữ liệu lớn.
 - **Structured Streaming:** Xử lý và áp dụng mô hình để dự đoán trên luồng dữ liệu **real-time**.
-  **Apache Kafka:** Hệ thống message queue, đóng vai trò trung gian nhận và phân phối luồng giao dịch đến hệ thống xử lý.
-  **Apache Airflow:** Công cụ điều phối (**orchestration**), tự động hóa việc khởi chạy và quản lý toàn bộ pipeline một cách tuần tự.
-  **Docker:** Đóng gói và triển khai nhất quán các thành phần của hệ thống trên một môi trường cluster nhiều máy.

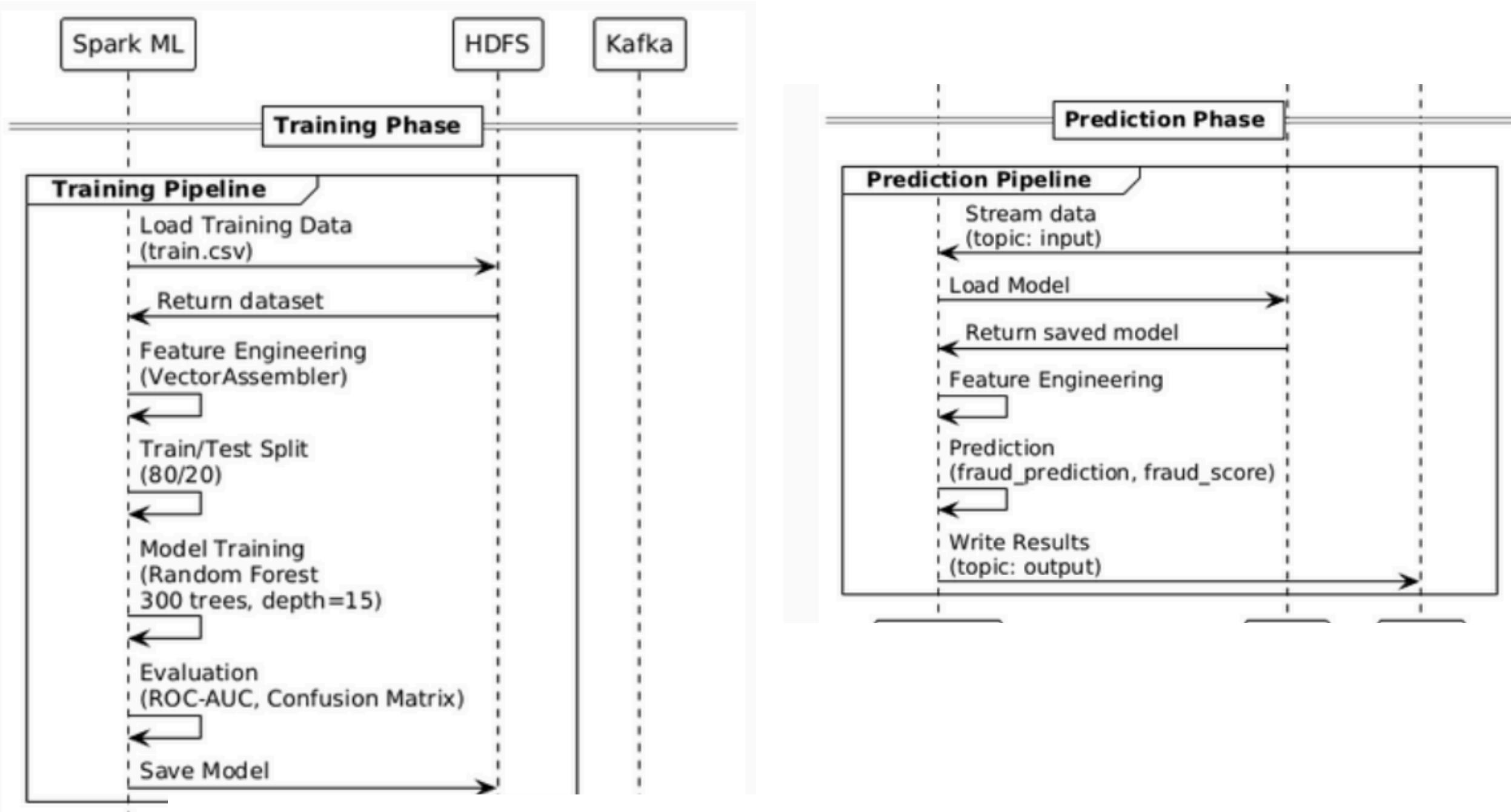


Kiến trúc Hệ thống Tổng thể (System Architecture)

Hình 3.1: Kiến trúc hệ thống tổng thể



Spark ML Pipeline & Metrics



Kết quả thực nghiệm

Kết quả trên hệ thống phân tán



0.98 ROC-AUC

Độ chính xác mô hình



< 1 giây

Độ trễ dự đoán

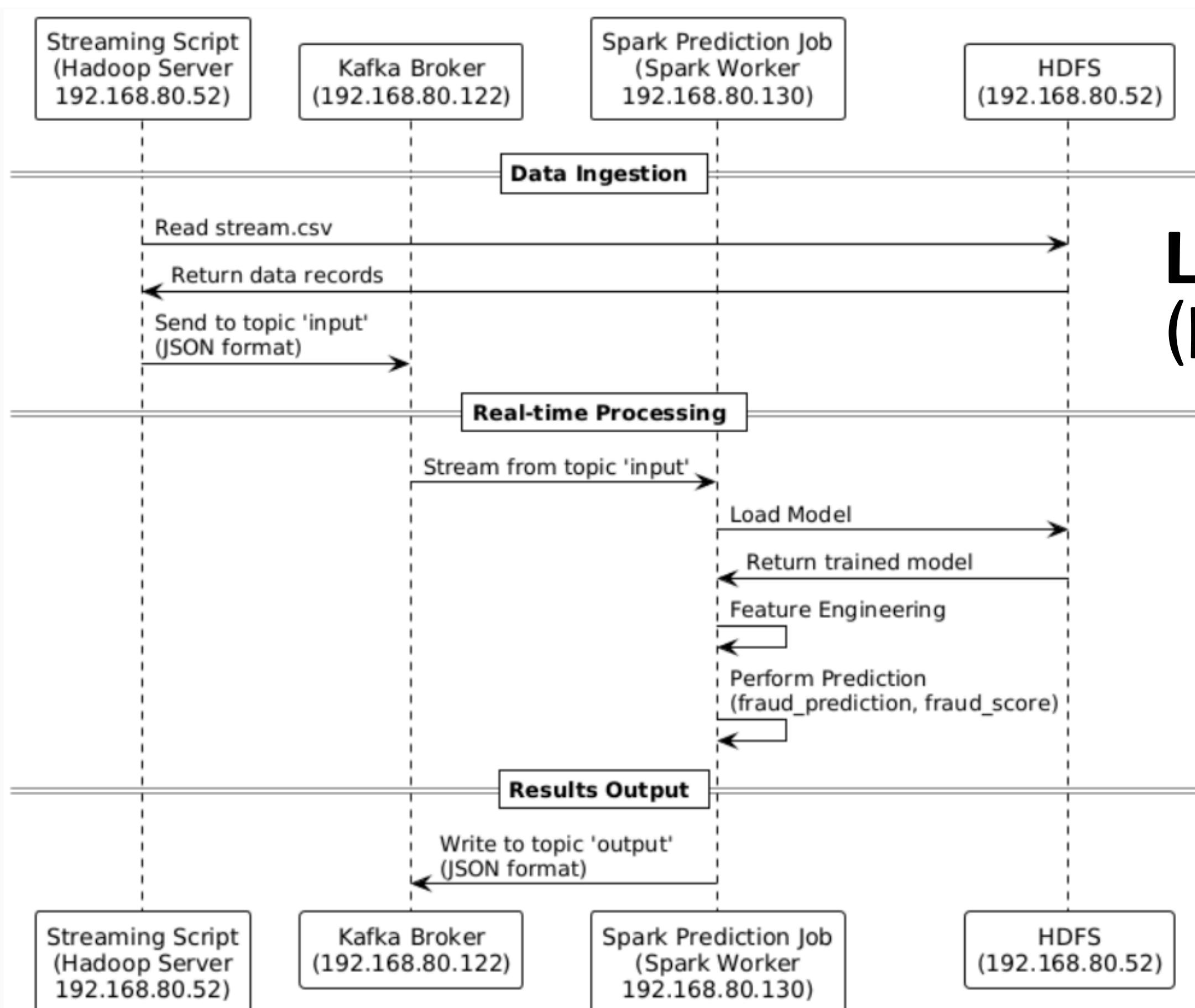


~15 phút

Thời gian huấn luyện
(trên toàn bộ dữ liệu)

Bảng 4.2: Các chỉ số hiệu suất của hệ thống

Hình 3.5: Spark ML Pipeline cho training và prediction



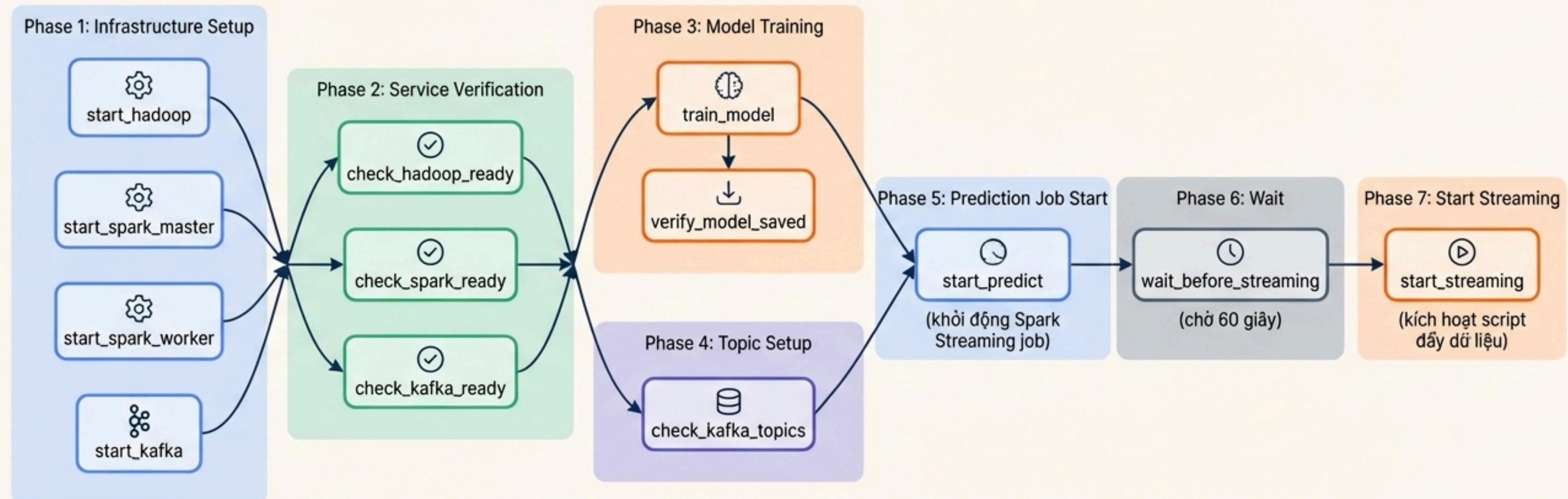
LUỒNG DỮ LIỆU CHI TIẾT (DATA FLOW)

Hình 3.4: Flow dữ liệu qua Kafka topics



Tự động hóa Phức hợp với Airflow DAG

Quy trình End-to-End được định nghĩa bằng code, đảm bảo tính lặp lại, tin cậy và dễ dàng giám sát.





Kế hoạch triển khai và phân công ban đầu

Phân chia công việc:

- **Phan Văn Tài:** Phụ trách pipeline streaming (Kafka, Spark Streaming) và hệ thống điều phối Airflow.
- **Phan Minh Thuy:** Phụ trách chuẩn bị dữ liệu, huấn luyện mô hình (Spark ML), và xây dựng giao diện người dùng (UI).

Timeline ban đầu:

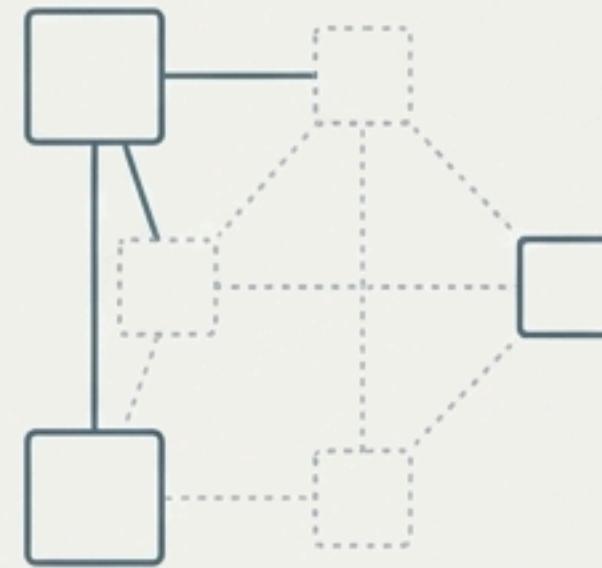
Dự kiến hoàn thành trong 1 tuần.

Các bước thực hiện dự kiến:

- 1 Chuẩn bị dữ liệu và lưu trữ trên HDFS.
- 2 Huấn luyện và lưu mô hình bằng Spark ML.
- 3 Xây dựng luồng dữ liệu streaming bằng Kafka.
- 4 Tích hợp Spark Streaming để đọc dữ liệu từ Kafka và thực hiện dự đoán.
- 5 Sử dụng Airflow để tự động hóa và điều phối toàn bộ hệ thống.



Thách thức trước khi bắt đầu: Áp lực và giới hạn



**** Áp lực thời gian cực lớn:**

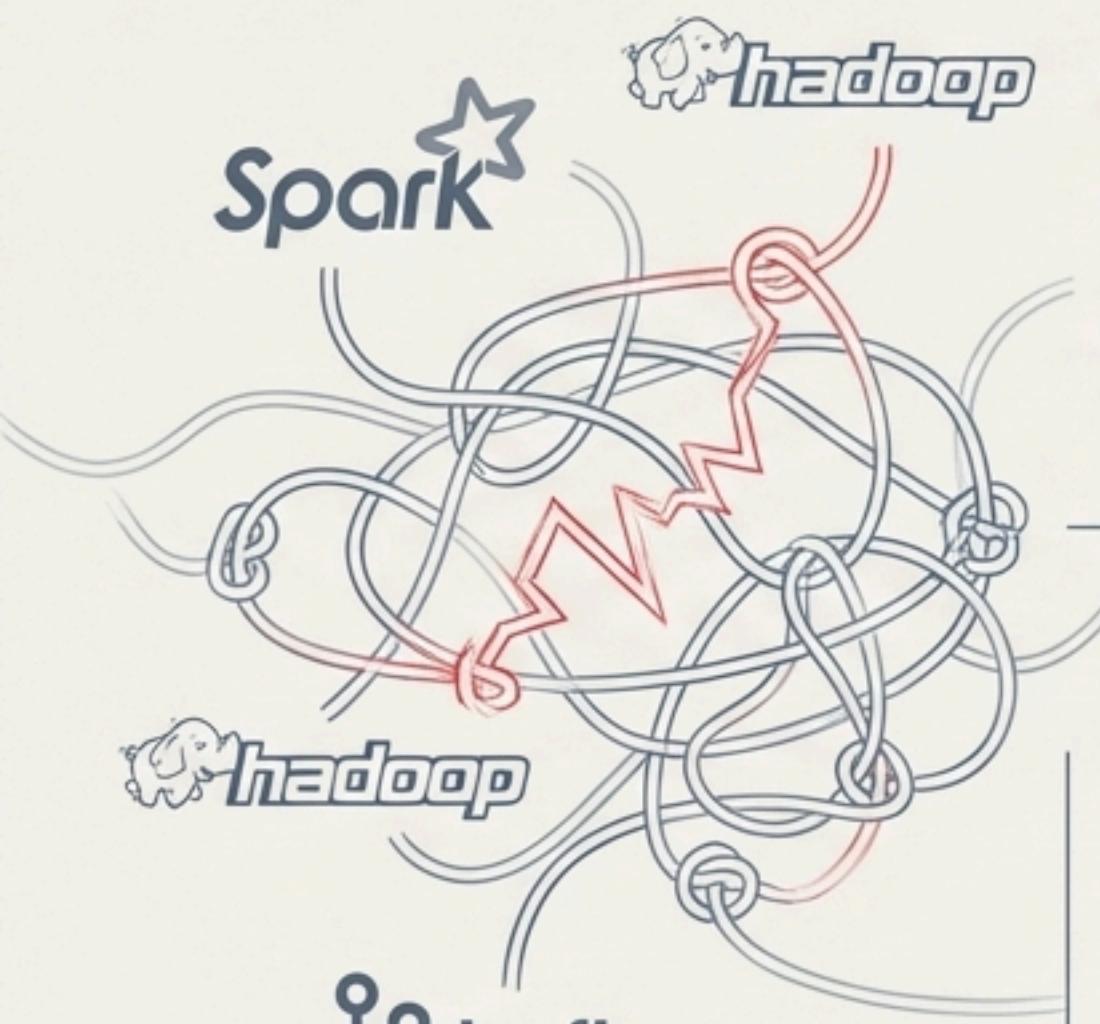
- Ban đầu, nhóm chỉ còn 2 ngày để hoàn thành toàn bộ dự án.
- Nhờ sự xem xét của giảng viên, nhóm được gia hạn thêm 5 ngày, đây là cơ hội quý giá để xây dựng một hệ thống chỉn chu.

**** Kiến thức và kinh nghiệm còn hạn chế:**

- Các công nghệ Big Data như Spark, Kafka, Airflow còn tương đối mới với cả nhóm.
- Thiếu kinh nghiệm thực tế trong việc cấu hình và triển khai một hệ thống phức tạp trên nhiều máy (multi-node cluster).



"Cơn ác mộng" trong quá trình tích hợp hệ thống

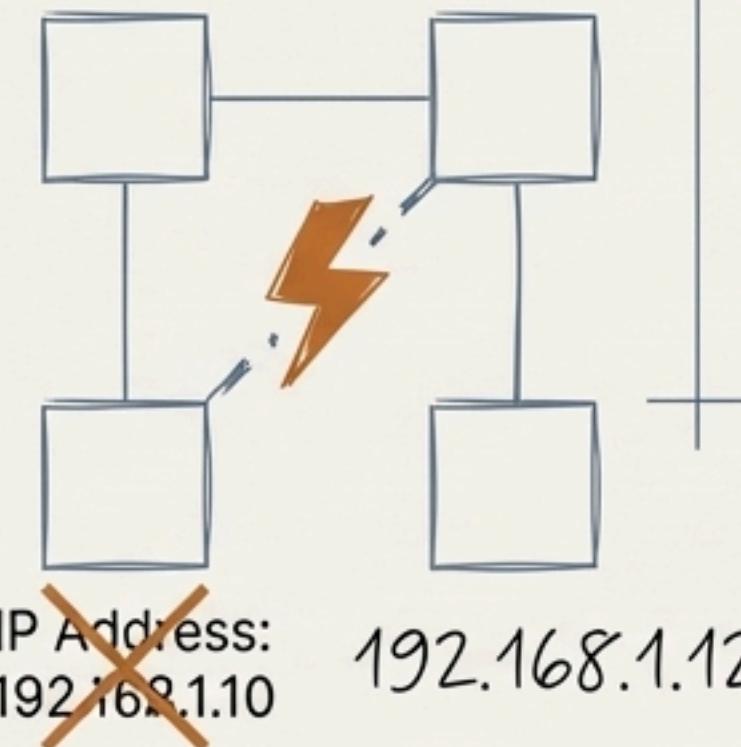


Xung đột phiên bản (Dependency Hell)

Mất rất nhiều thời gian để tìm ra một bộ phiên bản tương thích giữa Spark, Hadoop, Kafka, và các thư viện Python liên quan.

Sự cố hạ tầng không lường trước

Một máy trong cluster đột ngột thay đổi địa chỉ IP, làm sập toàn bộ kết nối giữa các node, gây ra lỗi hàng loạt và mất nhiều giờ để khắc phục.



Nhầm lẫn trong quản lý cluster

Gặp khó khăn trong việc xác định chính xác máy nào đang chạy Spark Master và máy nào là Worker.

Việc gỡ lỗi (debug) trên một hệ thống phân tán phức tạp và tốn thời gian hơn rất nhiều so với trên một máy đơn.

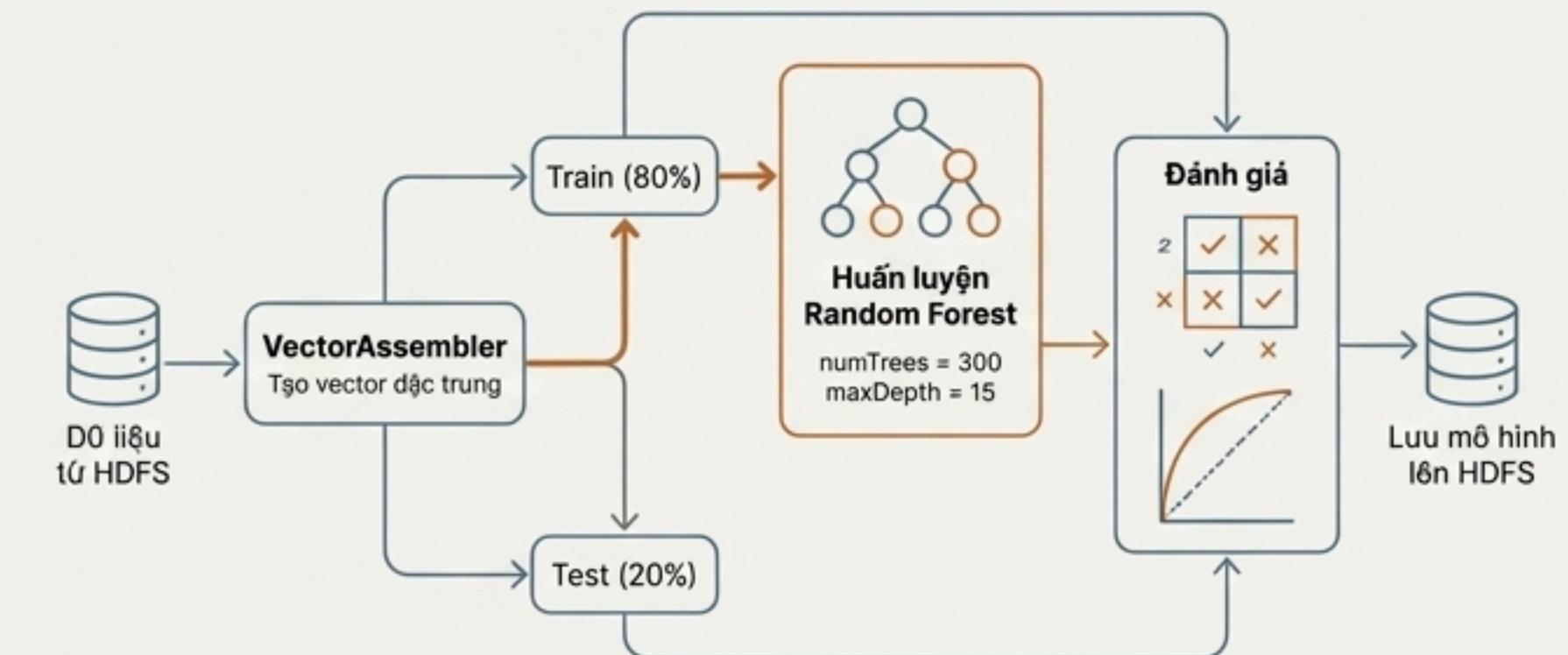
Hạn chế Kiến trúc: IP-based Queue vs. Redis

Giải pháp Hiện tại (IP-based Queue)	Giải pháp Tối ưu (Redis Queue)
Cấu hình cứng Hardcoded IP trong code Hậu quả: Khó mở rộng, dễ lỗi khi đổi IP	Linh hoạt Worker chỉ kết nối Redis Lợi ích: Scale dễ, không sửa code
No Persistence Mất task khi worker crash	Persistence Redis lưu task trên đĩa
Network Latency Hiệu suất phụ thuộc mạng	In-memory Processing Tốc độ nhanh, độ trễ thấp



Lõi của hệ thống: Huấn luyện mô hình Machine Learning

- **Bài toán:** Phân loại nhị phân (Binary Classification) để xác định một giao dịch là 'bình thường' (0) hay 'gian lận' (1).
- **Thuật toán sử dụng:** Random Forest Classifier trong thư viện Spark ML.
 - Đây là một mô hình ensemble mạnh, giúp giảm overfitting và cho hiệu năng tốt trên dữ liệu dạng bảng.
- **Quy trình huấn luyện trên Spark:** (Các bước được thể hiện trực quan trong sơ đồ bên phải)
- **Đánh giá:** Sử dụng các chỉ số quan trọng cho bài toán mất cân bằng như Confusion Matrix, Classification Report (Precision, Recall, F1-score) và ROC-AUC.

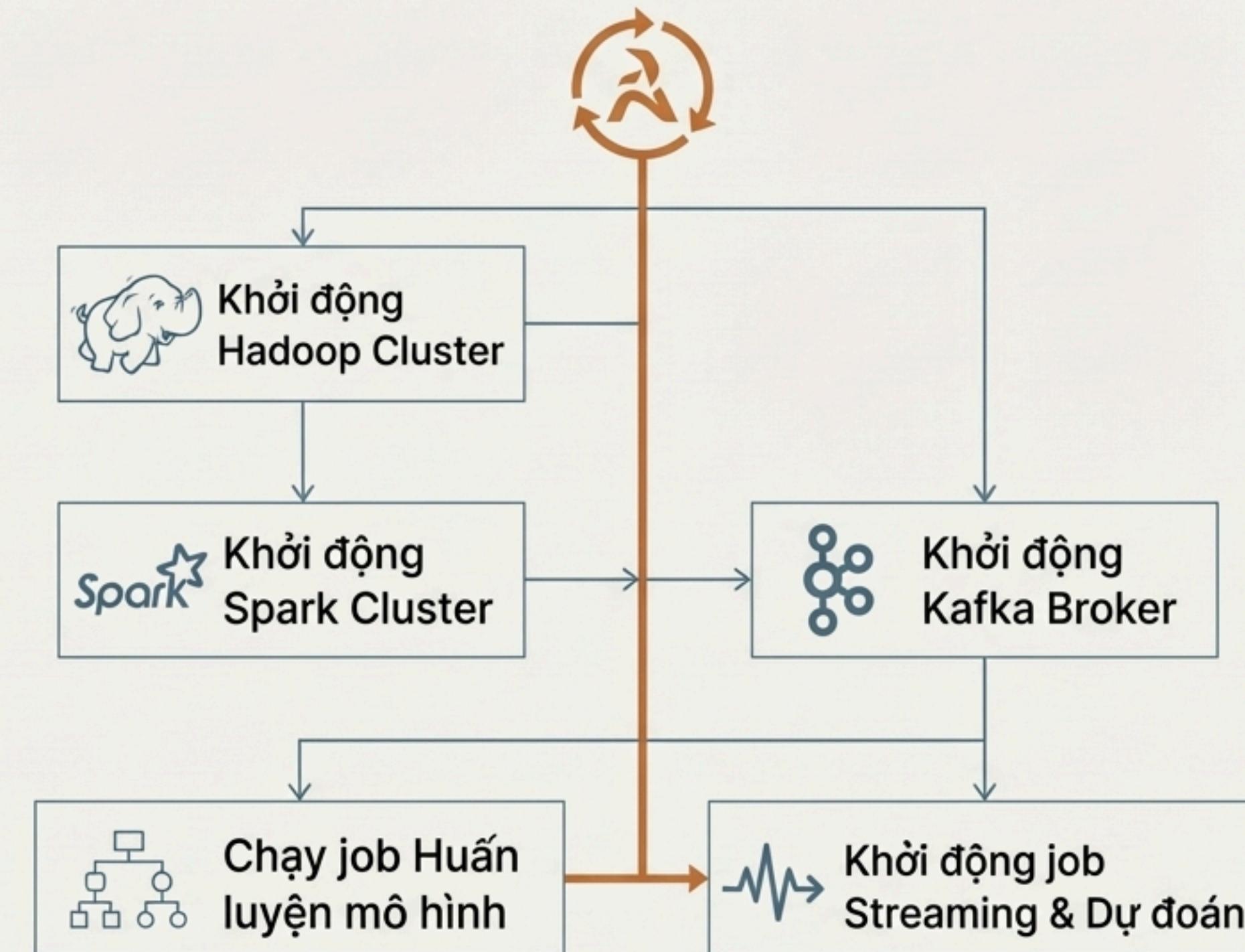




Airflow - "Nhạc trưởng" điều phối toàn bộ hệ thống

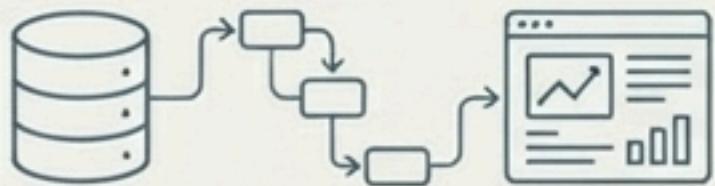
Vai trò: Biến một quy trình khởi chạy thủ công, phức tạp và dễ lỗi thành một workflow hoàn toàn tự động, tin cậy và có thể giám sát được.

Lợi ích: Tự động hóa hoàn toàn việc quản lý cluster, đảm bảo hệ thống vận hành ổn định và giảm thiểu sai sót do con người.





Thành tựu: Một hệ thống Big Data hoàn chỉnh và hiện đại



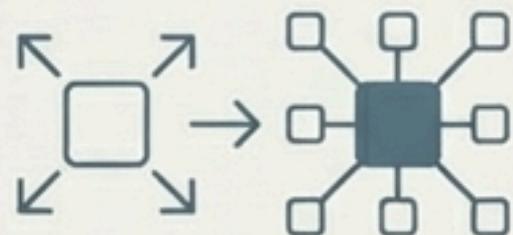
- **Pipeline End-to-End:** Xây dựng thành công một quy trình hoàn chỉnh từ lưu trữ dữ liệu thô đến dự đoán real-time và hiển thị UI.



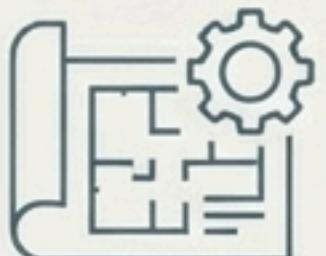
- **Kiến trúc Hybrid:** Kết hợp linh hoạt giữa xử lý theo lô (batch processing) để huấn luyện mô hình và xử lý luồng (streaming) để dự đoán.



- **Tự động hóa toàn diện:** Toàn bộ hệ thống được điều phối tự động bằng Apache Airflow, giảm thiểu can thiệp thủ công và tăng độ tin cậy.



- **Khả năng mở rộng (Scalability):** Kiến trúc được thiết kế để dễ dàng thêm các node mới vào cluster Spark/Hadoop nhằm tăng năng lực xử lý.



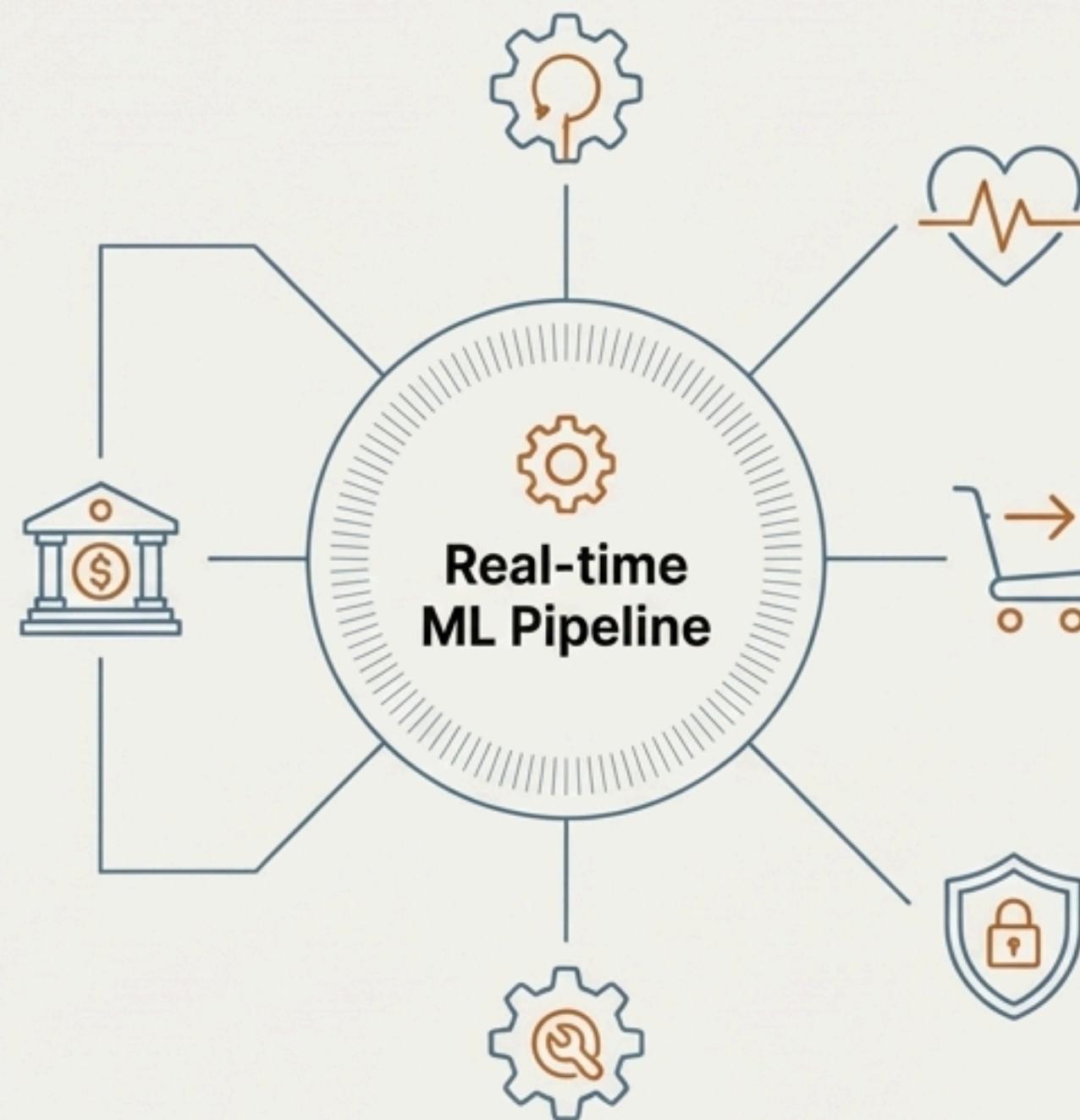
- **Mô phỏng sát với thực tế:** Quy trình và công nghệ được sử dụng rất gần với các hệ thống Big Data đang vận hành tại các doanh nghiệp lớn.



Khả năng ứng dụng và mở rộng trong thực tế

Ứng dụng trực tiếp trong ngành tài chính

- Xây dựng hệ thống phát hiện gian lận cho ngân hàng, công ty tài chính, ví điện tử.
- Giám sát các giao dịch chứng khoán đáng ngờ trong thời gian thực.



Tiềm năng mở rộng sang các lĩnh vực khác

- **Phát hiện bất thường** (Anomaly Detection): Giám sát các chỉ số vận hành của server hoặc thiết bị IoT để cảnh báo sự cố.
- **Thương mại điện tử**: Phát hiện các đơn hàng lừa đảo hoặc hành vi lạm dụng khuyến mãi.
- **An ninh mạng**: Phân tích log hệ thống real-time để phát hiện các dấu hiệu của một cuộc tấn công mạng.



Tổng kết, bài học kinh nghiệm và định hướng tương lai



Những gì đã làm được

- Hoàn thành một hệ thống Big Data phức tạp, tích hợp thành công Spark, Kafka và Airflow trên môi trường multi-node.
- Xây dựng được pipeline streaming real-time từ khâu nhận dữ liệu đến dự đoán và hiển thị kết quả.



Bài học kinh nghiệm quan trọng

- Làm việc với hệ thống phân tán đòi hỏi sự kiên nhẫn và phương pháp gỡ lỗi có hệ thống.
- Tự động hóa (Orchestration) không phải là một tùy chọn, mà là yêu cầu bắt buộc để quản lý các hệ thống phức tạp một cách hiệu quả.



Hướng phát triển tiếp theo

- **Cải thiện mô hình:** Áp dụng các kỹ thuật xử lý dữ liệu mất cân bằng (SMOTE) và thử nghiệm các thuật toán khác như XGBoost.
- **Triển khai MLOps:** Tích hợp MLflow để quản lý vòng đời mô hình và Prometheus/Grafana để giám sát hệ thống chuyên nghiệp.

Lời cảm ơn: Nhóm xin chân thành cảm ơn thầy và các bạn đã lắng nghe.

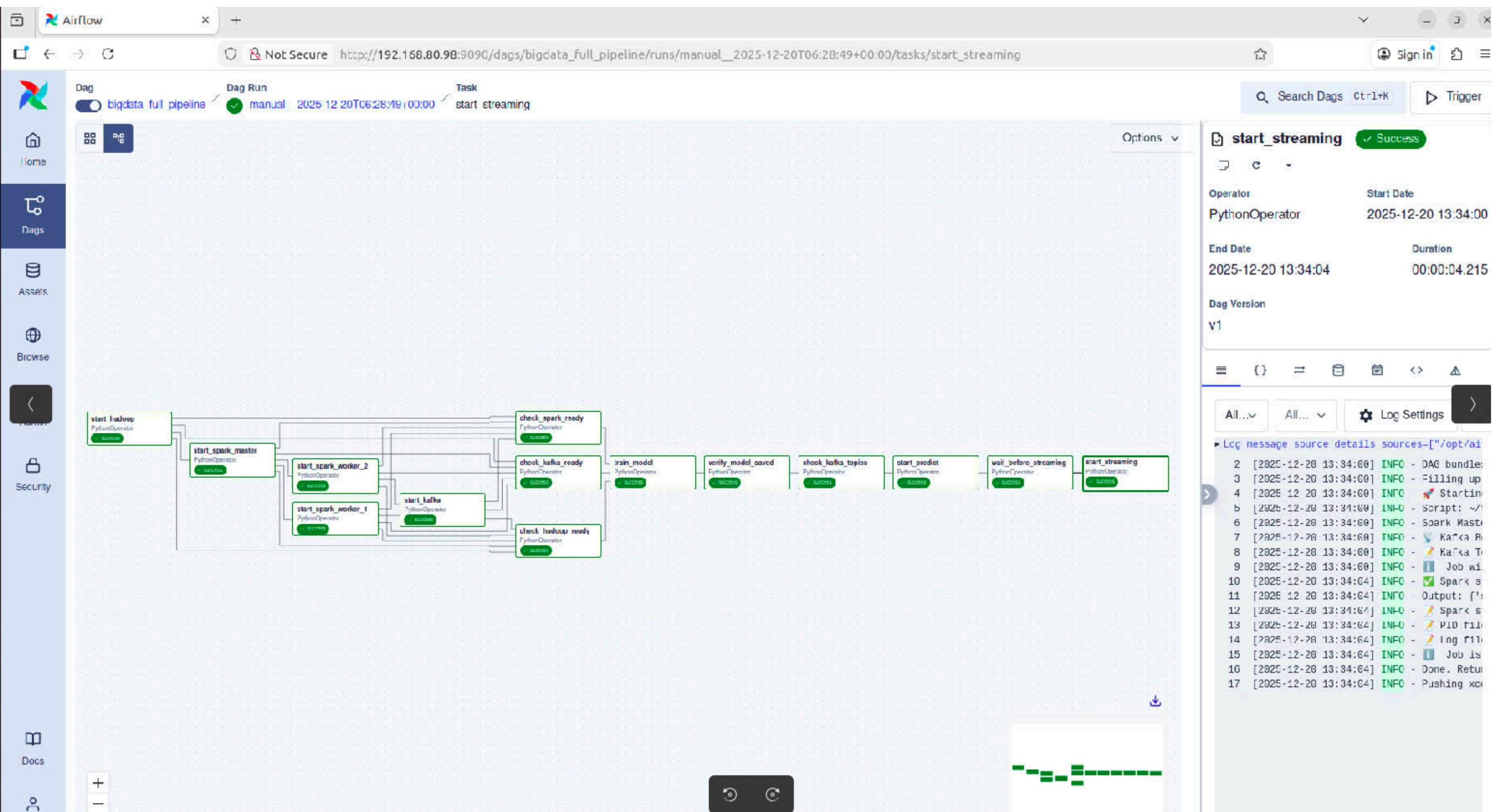


Nhóm 2

DEMO

Mã nguồn dự án

https://github.com/ANGFLO26/finalproject_bigdata.git





THANK YOU!

Our group would like to express our deepest gratitude to you, Professor, and wish you continued good health, happiness, and inspiration for future generations of students.