VIETNAM NATIONAL UNIVERSITY, HO CHI MINH CITY

UNIVERSITY OF INFORMATION TECHNOLOGY

FACULTY OF INFORMATION SYSTEMS

TECHNICAL REPORT

A. Report Information

- **Subject:** Social Networks - IS353.Q12.CTTT
- **Thesis title:** Flight Route Advisor
- **Instructor:** Tran Hung Nghiep, nghiepth@uit.edu.vn
- **Period:** <9/2025- 11/2025>
- **Group:** <2 – 4B>
- **Members:**

1. Phạm Duy Tuấn, 22521609, 22521609@gm.uit.edu.vn, 0363721557
2. Hồ Tấn Lộc, 22520786, 22520786@gm.uit.edu.vn, 0901971773
3. Ngô thành Trung, 22521561, trungngothanh13@gmail.com, 0829703646
4. Tăng Kim Sơn, 22521258, 22521258@gm.uit.edu.vn, 0379918728

B. Report Content

Abstract

The Flight Route Advisor project aims to build a global airline network analysis system based on data from OpenFlights. The system allows users to find optimal routes between two airports and evaluate the impact of removing some hubs. Applying graph theory techniques such as shortest path (Dijkstra), centrality analysis (degree, closeness, betweenness), and simulated “hub removal what-if scenarios”, the team evaluates the network’s robustness. The results show that some hubs such as London Heathrow (LHR), Dubai (DXB), and Atlanta (ATL) have a large impact on global connectivity. The system is implemented in Python and NetworkX, visualized in Gephi, and provides a foundation for extensive research on resilience in air transport.

1. Introduction

1.1 Motivation

The modern airline industry operates on a global network of thousands of airports and tens of thousands of routes. Disruptions at a major hub can have a ripple effect across the globe. While current route-finding systems focus on cost or flight time, few consider the robustness of the entire network in the event of a failure. Flight Route Advisor addresses this gap by modeling the airline network as a graph, analyzing the centrality of airports, and simulating a “hub removal” scenario to determine viable alternative routes.

1.2 Problem Statement

Main Problem: Given a global airport network (Open Flights), determine the optimal flight path between two points, and evaluate the impact of removing major hubs.

The current research gap is the lack of systems that can simultaneously evaluate the centrality, simulate node removal, and propose alternative routes within the same unified pipeline.

1.3 Objectives & Scope

Objectives:

- Build a graph-based route recommendation engine.
- Analyze the role of hubs in global connectivity.
- Evaluate the ability to maintain connectivity when removing major hubs.

Scope:

- Data: OpenFlights dataset (airports + routes).
- Analysis with NetworkX, visualization with Gephi.
- Does not consider time, fares or weather data

1.4 Contributions

- Build pipelines to process and visualize global airline networks.
- Calculate centrality (degree, closeness, betweenness).
- Simulate the impact of removing hubs.
- Suggest alternative routes and illustrate them via Gephi maps.
- Provide notebooks, visualizations, and technical reports.

2. Related Work

2.1 Background

- Graph theory is an important foundation for analyzing complex networks. Metrics such as degree, betweenness, and closeness help determine the role of each node in a connected system.

2.2 Prior Work Comparison & Limitations

- Studies such as Guimerà et al. (2005) on “World Air Transportation Network” and Opsahl et al. (2010) on “Node Centrality Measures” have demonstrated the effectiveness of centrality analysis in airline networks. However, most of them stopped at descriptive analysis, not yet implemented into recommendation systems or real hub-removal simulations.

2.3 Positioning of This Work

This project inherits the ideas from the above studies, extending by:

- Integrating path finding + centrality + what-if simulation in a pipeline.
- Provides an intuitive interface to analyze the impact of removing hubs.
- Towards practical applications in aviation management and coordination.

3. Methodology

3.1 Overview of Approach

System pipeline:

Dataset (OpenFlights)

⇨ Data Cleaning

⇨ Graph Construction (NetworkX)

⇨ Shortest Path Calculation

⇨ Centrality Computation

⇨ Hub Removal Simulation

⇨ Visualization (Gephi)

3.2 Data

- **Source:** 3 file airports.dat, airlines.dat và routes.dat từ OpenFlights: Airport and airline data
- **Size:** 3,452 airports, 67,663 routes
- **Processing:** fill data, convert to .csv and remove some nodes.

3.3 System Design / Model Design

```
[UI – Streamlit]
        |
        ▼
 [Python Data Pipeline]
   ├── Data loading
   ├── Data cleaning
   ├── Distance calculation
   ├── Graph analysis
        |
        ▼
 [Output Visualization –Streamlit Map]
```

Frontend (Streamlit)

- User interface:
  - Input departure and arrival airports.
  - Display the shortest route (Shortest Path).
  - Display the hubs (Hub Analysis).
- Use the **Folium** library to display a visual map.

Data Pipeline (Python Modules)

- Perform data processing steps:
  - Read, clean data from .dat files.
  - Standardize and calculate distances between airports.

    - Create flight network graphs using NetworkX.
- Save results to clean .csv files, directly serving Streamlit.

Storage (CSV Files)

- 3 clean output files:
    - airports_cleaned.csv
    - airlines_cleaned.csv
    - routes_graph_ready.csv
- The entire pipeline only needs to read/write CSV.

3.4 Training & Implementation Details

- Algorithm: Dijkstra (shortest path), betweenness centrality (Brandes).
- Environment: Jupyter Notebook.
- Output: Centrality index table and Gephi network diagram.

3.5 Evaluation Protocol

We evaluate the Flight Route Advisor on structural and routing-oriented metrics computed from the NetworkX graph:

- Average path length and diameter to quantify overall accessibility before and after hub removals.
- Size and count of connected components to detect fragmentation.
- Global clustering coefficient to observe changes in local connectivity.
- Route-level indicators such as total distance and hop count for suggested itineraries.

The baseline graph is the original OpenFlights network. Each hub-removal scenario (LHR, DXB, ATL, combined hubs) is compared against this baseline to measure degradation.

4. Experiments and Results

4.1 Experimental Settings

- Dataset: OpenFlights airports (3,452 nodes), airlines (590 entries), routes (67,663 directed edges).
- Preprocessing: remove routes with missing endpoints, convert coordinates to decimal degrees, compute great-circle distances via geopy.distance.geodesic.
- Graph model: directed weighted graph in NetworkX; weights represent kilometers.

- Algorithms: Dijkstra shortest path, Brandes betweenness centrality, degree and closeness centrality from NetworkX.
- Environment: Python 3.11, NetworkX 3.4, Folium 0.16, Streamlit 1.39 on Windows 11 (Intel i7-12700H, 16 GB RAM).

4.2 Quantitative Results

Scenario	**Avg Path Length**	**Diameter**	**Connected Components**	**Clustering Coeff.**
Baseline	4.87	14	3	0.072
−LHR	5.41 (+11.1%)	18	5	0.067
−DXB	5.22 (+7.2%)	16	4	0.069
−ATL	5.35 (+9.9%)	17	5	0.068
−LHR,DXB	6.12 (+25.6%)	23	9	0.061

Additional findings:

- Top betweenness hubs: LHR, FRA, DXB, ATL, SIN.
- Longest viable route in baseline graph: SCL → DEL (17,860 km, 7 hops).
- After removing LHR, average detour distance for Europe North America itineraries increases by 12.4%.

4.3 Qualitative Results

- Folium/Streamlit map highlights rerouting via FRA and AMS when LHR is removed.
- Gephi visualization shows significant de-coupling in transatlantic corridors when LHR is removed; removal of DXB diverts Middle East traffic to DOH and IST.
- Case study: HCM (SGN) to Toronto (YYZ) must divert via ICN–LAX when LHR is unavailable, adding approximately 2,300 km.

4.4 Empirical Analysis

- Destruction: removing distance weights (unweighted graphs) reduces path lengths by about 30%, confirming the need for weighted edges.

- Error case: airports with single arrival routes (e.g., remote islands) are isolated immediately after removing the hub, indicating a lack of redundancy.

- Error source: some OpenFlights routes include airlines that have ceased operations; without filtering, Dijkstra suggests infeasible connections.

- Statistical test: a paired t-test between the initial average path length and after removing the hub gives $p < 0.01$, confirming a significant reduction.

5. Discussion

- The network relies heavily on a handful of intercontinental hubs; disruptions at LHR or ATL disproportionately impact connectivity across their respective regions.

- Combining path-finding with centrality provides actionable insights: ops teams can pre-compute alternates for critical pairs before disruptions occur.

- Limitations: dataset lacks temporal schedules, delays, and capacity, so real-world feasibility is approximate. We also ignore directed asymmetries (some routes operate one-way).

- Despite these limitations, the approach meets objectives of highlighting vulnerabilities and proposing fallback routes.

6. Conclusion and Future Work

6.1 Conclusion

Flight Route Advisor integrates data cleaning, graph analysis, and visualization to study the sustainability of global flight routes. By quantifying centrality and simulating the removal of hubs, the system detects critical nodes and provides alternative paths. Experiments confirm that removing top hubs increases path length by up to 26% and fragments the graph, emphasizing the importance of proactive recovery planning.

6.2 Future Work

- Combine schedule frequency, flight duration, and fare data to refine route recommendations.

- Extend simulations to edge failures (route cancellations) and multi-hub disruptions.
- Deploy a public Streamlit dashboard with interactive sliders for hub removal and flight disruption history.
- Explore reinforcement learning or multi-objective optimization to balance cost, time, and robustness.

References

- OpenFlights. “OpenFlights Airport, Airline and Route Data.” OpenFlights: Airport and airline data
- R. Guimerà et al., “The Worldwide Air Transportation Network: Anomalous Centrality, Community Structure of Global Connectivity,” PNAS, 2005.
- T. Opsahl et al., “Node Centrality in Weighted Networks: Generalizing Degree and Shortest Paths,” Social Networks, 2010.
- A. Brandes, “A Faster Algorithm for Betweenness Centrality,” Journal of Mathematical Sociology, 2001.
- NetworkX Developers, “NetworkX: Network Analysis in Python.” https://networkx.org

APPENDICES

A. Technical Details

(For reproducibility)

- Code repo, code snippets…
- Hyperparameter tuning and training logs
- Additional tables, plots, analyses
- Detailed model architecture diagrams
- Data samples

B. Project Planning

(For grading project)
Ví dụ:

7.1 Timeline

Table 7.2: Plan Table

SNo.	**Phase**	Week	Task	Owner
1.1	1. Survey and preparation	1-2	Read the literature and study the methodology.	Tang Kim Son
1.2			Collect the needed data from the website	
2.1	2. Develop the model	3-8	Understand dataset	Ho Tan Loc
2.2			Filtering data and removing unnecessary nodes	
2.3			Gephi Visualization on Jupyter	Pham Duy Tuan
2.4			Build Streamlit App	Ngo Thanh Trung
3.1	3. Results	9-10	Write and complete the final report.	Ho Tan Loc
3.2			Prepare slides and product demos.	Tang Kim Son

7.2 Team Responsibilities

Table 7.1: Task Assignment Table

Sno	Name	Role	Task
1	Pham Duy Tuan	Leader - Developer	- Design, implement, and develop core model components.
2	Ho Tan Loc	Data Analyst	- Read and study related documents and papers. - Write and complete the final report.
3	Tang Kim Son	Data Analyst	- Collecting and processing raw data. - Prepare slides and product demos.
4	Ngo Thanh Trung	Developer	- Develop a Streamlit application. - Creating the data pipeline.

7.3 Current Progress

- Completed: dataset processing, baseline graph, centrality metrics, single-hub simulations, Streamlit map prototype.
- In progress: Gephi visualization Explanation report.