SmartTransit - A Data-Driven System for Navigating and Analyzing Public Transport

Mitul Nakrani, Priyanka Maan

Abstract

This report presents a comprehensive technical and methodological overview of the SmartTransit interactive web dashboard for navigating and analyzing Boston's MBTA subway network. Developed in Python, the system leverages Streamlit for the user interface, Pandas and NumPy for data processing, NetworkX for graph-based analytics, and embedded HTML/CSS for enhanced visual styling. By integrating GTFS transit datasets and additional public transport data, the application enables users to filter, query, and explore transportation insights through a browser-based interface. Key features include AI-assisted natural language querying powered by GPT-4, route and travel time analysis, accessibility mapping, and interactive visualizations with Plotly.

1 Introduction

Public transportation plays a pivotal role in shaping urban mobility, reducing traffic congestion, and minimizing environmental impact. In Boston, the Massachusetts Bay Transportation Authority (MBTA) subway network serves as a lifeline for daily commuters, handling approximately 425,000 weekday station entries and 200,000 on weekends. However, the wealth of transit data produced daily often remains inaccessible or unintuitive for city planners, commuters, and researchers.

The SmartTransit project bridges this gap by offering a comprehensive, data-driven platform for analyzing and visualizing Boston's Red, Green, Orange, and Blue subway lines. Combining interactive dashboards and AI-powered natural language querying, the system empowers users to gain actionable insights into the network's performance, accessibility, and operational patterns.

2 Literature Review

Open data, network analysis, and AI are increasingly used to improve public transit planning and operations. The General Transit Feed Specification (GTFS) [1] is the global standard for sharing transit schedules and routes, enabling consistent integration of datasets from agencies such as the MBTA [2] and MassDOT. Research has shown that GTFS data can support service optimization and comparative performance evaluations.

Graph theory provides a useful framework for modeling transit systems, where stations are nodes and routes are edges. Metrics such as centrality and shortest paths help identify critical hubs and bottlenecks, as demonstrated by Derrible and Kennedy [3]. Tools like NetworkX make these analyses accessible and computationally efficient.

Natural language to SQL systems using OpenAI [7] allow non-technical users to explore complex datasets without advanced technical skills. In transit contexts, this can support decision-making by quickly answering operational and planning questions.

Finally, visualization tools such as Plotly [5] and Streamlit [6] enable interactive dashboards for exploring spatial, temporal, and performance trends. Studies like Ferreira et al. [4] highlight the importance of geospatial mapping and trend analysis for effective transport planning. These insights form the base of SmartTransit's design, combining open data, graph analytics, AI, and interactive visualization into a unified platform.

3 Project Scope

The SmartTransit project is designed to analyze, visualize, and enhance the accessibility of the MBTA subway network through the integration of open transit data, graph analytics, and AI-assisted querying. The scope of the project is defined as follows:

- **Geographical Coverage:** The current implementation focuses exclusively on the Massachusetts Bay Transportation Authority (MBTA) subway system, including the Red, Green, Orange, and Blue Lines.
- **Data Sources:** The platform utilizes publicly available GTFS feeds from the MBTA, supplemented by datasets from the Massachusetts Department of Transportation (MassDOT), with a focus on stops, routes, travel times, and accessibility facilities.
- **Functional Capabilities:** Core functions include travel time analysis, accessibility mapping, route performance comparisons, AI-assisted natural language querying, and interactive visualization through a web-based dashboard.

This scope ensures a focused and achievable implementation while establishing a foundation for future enhancements, including the integration of real-time data feeds, predictive models, and additional transportation modes.

4 Methodology

The SmartTransit system employs a structured methodology designed to ensure the accurate collection, preprocessing, analysis, and visualization of MBTA transit data. The process begins with comprehensive data acquisition from official and supplementary sources, followed by meticulous preprocessing and organization into structured formats suitable for analysis. Subsequently, the data is stored in an optimized database environment, enabling efficient statistical computation and advanced queries. The methodology concludes with a combination of quantitative analysis and interactive visualizations to derive actionable insights.

4.1 Data Collection

Data for the SmartTransit system is primarily sourced from the official MBTA General Transit Feed Specification (GTFS) feed, which provides industry-standard information on stops, routes, schedules, and facilities. To enhance the dataset, supplementary travel

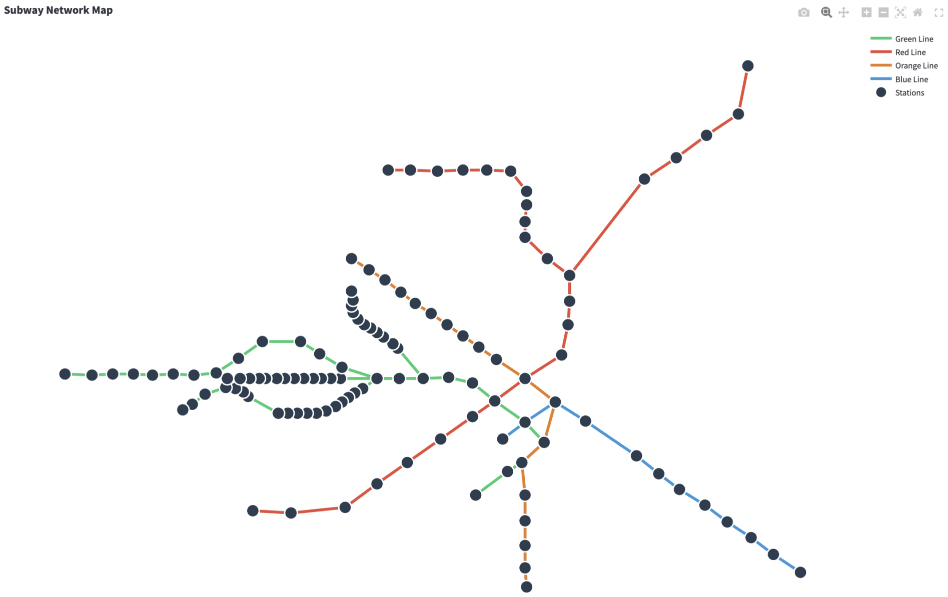


Figure 1: Boston's MBTA Network Map

time information is incorporated from the Massachusetts Department of Transportation (MassDOT) open data directory. The GTFS format, widely used across the public transportation sector, ensures compatibility and consistency, facilitating integration with analytical and visualization tools. This dual-source approach allows for both comprehensive coverage and cross-verification of transit data.

4.2 Data Preprocessing

The preprocessing stage begins by fetching the latest GTFS data directly from MBTA's official APIs and the MassDOT data repositories. From these feeds, only the most relevant files are extracted, including `stops.txt`, `routes.txt`, `travel_times.csv`, `facilities.txt`, and `connections.csv`. The GTFS data is subsequently converted into CSV format to enhance compatibility with analytical workflows. A structured transformation process is applied to convert raw, unstructured datasets into relational tables, such as `stations`, `connections`, `locations`, and `station_amenities`. This step involves handling missing values, standardizing field formats, and reconciling discrepancies between data sources to ensure high-quality inputs for further analysis.

4.3 Data Storage

Following preprocessing, the cleaned and structured datasets are stored in an in-memory SQLite database. This design choice provides rapid query execution, reduces disk I/O latency, and supports real-time analysis scenarios. The database architecture supports SQL-driven analytical operations, enabling both statistical computation and the integration of AI-assisted query mechanisms. This setup allows for scalable exploration of the dataset, facilitating quick iteration on analysis workflows without compromising performance.

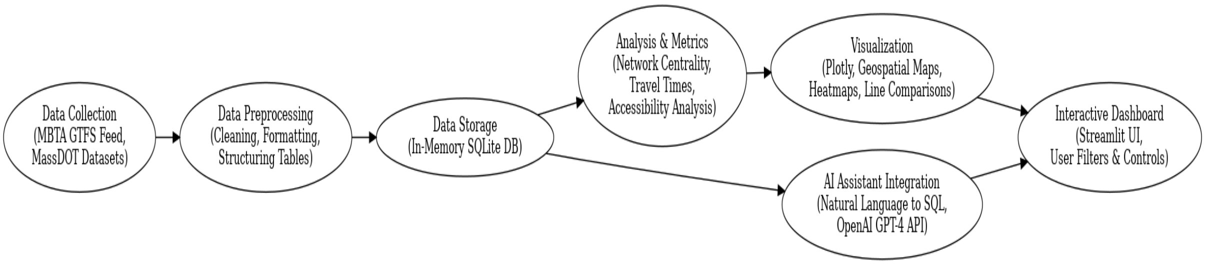

Data Collection (MBTA GTFS Feed, MassDOT Datasets)
Data Preprocessing (Cleaning, Formatting, Structuring Tables)
Data Storage (In-Memory SQLite DB)
Analysis & Metrics (Network Centrality, Travel Times, Accessibility Analysis)
Visualization (Plotly, Geospatial Maps, Heatmaps, Line Comparisons)
AI Assistant Integration (Natural Language to SQL, OpenAI GPT-4 API)
Interactive Dashboard (Streamlit UI, User Filters & Controls)

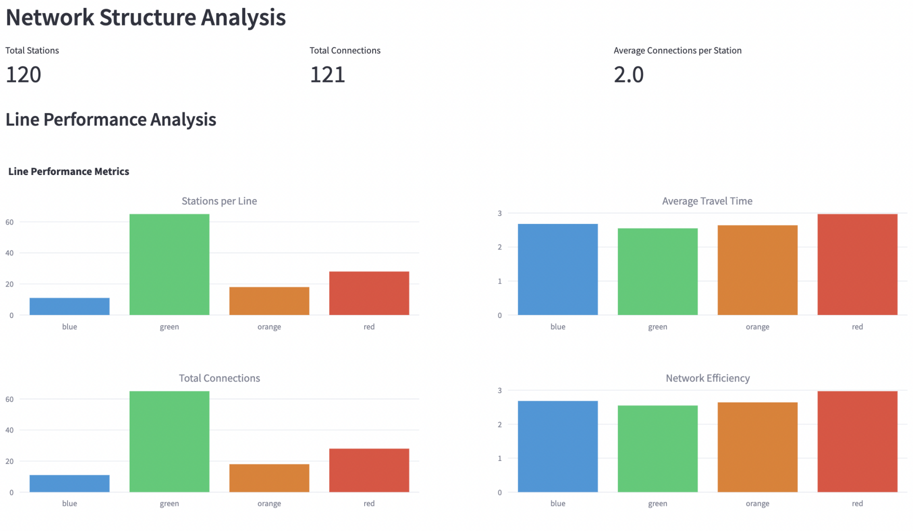
Figure 2: System Architecture

4.4 Analysis and Metrics

The analytical phase focuses on quantifying network performance and identifying operational bottlenecks. Key metrics include network centrality measures to determine the most critical transit hubs, travel time assessments to evaluate connection efficiency, and accessibility analysis to review the availability of elevators, ramps, and wheelchair access across stations. These metrics not only support operational decision-making but also provide a foundation for long-term infrastructure planning and service optimization.

4.5 Visualization

Visualization plays a crucial role in translating analytical findings into intuitive, actionable insights. Multiple visualization techniques are employed, including network graphs and heatmaps for understanding connectivity patterns, comparative line performance charts for cross-line evaluations, and geospatial scatter plots to reveal spatial distribution trends. In addition, interactive dashboards developed using Streamlit enable real-time exploration of the transit data, allowing users to filter, sort, and drill down into specific metrics. This interactive component enhances decision-making by providing a dynamic, user-driven view of the MBTA network's performance and accessibility.



Network Structure Analysis

Total Stations 120

Total Connections 121

Average Connections per Station 2.0

Line Performance Analysis

Figure 3: MBTA Network Visualizations

5 Technology

The SmartTransit project leverages a diverse set of modern tools, frameworks, and programming libraries to enable robust data processing, interactive visualization, and AI-assisted analytics. The choice of technology was driven by the need for scalability, real-time responsiveness, and ease of integration between different system components.

The frontend layer of SmartTransit is implemented using Streamlit, a Python-based web application framework well-suited for building interactive dashboards and data-driven interfaces. Streamlit's integration with HTML/CSS allows for custom styling, while the use of Plotly ensures that the visualizations are interactive and responsive.

The backend is built primarily in Python, taking advantage of its rich data science ecosystem. Libraries such as Pandas and NumPy are used for data manipulation and numerical computations, while SciPy supports advanced statistical analyses. Network analysis tasks, such as computing centrality and detecting bottlenecks, are handled by the NetworkX library, which offers efficient algorithms for graph-based analytics.

For data storage, an in-memory SQLite database is employed to provide fast query execution without the overhead of disk-based operations. This choice is particularly advantageous for iterative data exploration and AI-assisted querying. The data visualization layer relies heavily on Plotly, enabling the creation of interactive charts, geospatial maps, and comparative line performance plots.

AI integration is achieved through the OpenAI GPT-4 API, which enables natural language to SQL translation, allowing non-technical users to query the database using plain English. This feature enhances accessibility and broadens the range of stakeholders who can interact with the system without requiring deep technical knowledge.

Data collection is automated through direct access to the official GTFS feed from the MBTA, supplemented by datasets from the Massachusetts Department of Transportation (MassDOT). Once collected, the data undergoes preprocessing and transformation using Pandas and NumPy, ensuring that it is clean, structured, and ready for analysis.

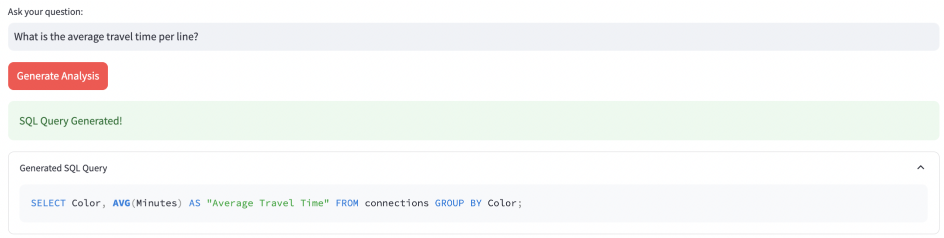
Table 1 summarizes the core technologies employed in the SmartTransit project, highlighting the tools and frameworks that underpin each component of the system.

Component	**Technology Used**
Frontend	Streamlit, HTML/CSS, Plotly
Backend	Python (Pandas, NumPy, SciPy, NetworkX)
Database	SQLite (in-memory)
Visualization	Plotly for interactive charts and maps
AI Integration	OpenAI GPT-4 API for natural language to SQL queries
Data Collection	GTFS official feed, MassDOT datasets
Data Processing	Pandas, NumPy for transformations and cleaning
Analytics	NetworkX for graph analysis, SciPy for statistical metrics

Table 1: Technologies used in the SmartTransit project

6 AI Assistant Integration

A key feature of the SmartTransit platform is its AI-powered natural language interface, which enables users to query the underlying transit database without requiring knowledge of SQL or data structures. This component leverages the OpenAI GPT-4 API to translate user-entered questions into optimized SQL statements, execute them against the in-memory SQLite database, and return results in both textual and visual formats. The AI assistant supports a wide range of query types, from accessibility checks (e.g., “Which stations on the Orange Line have elevators?”) to travel time requests (e.g., “What is the travel time from Harvard to Park Street?”). The integration is designed to handle ambiguous inputs by prompting for clarification, ensuring accurate results. Visual outputs are generated using Plotly, allowing for interactive charts, maps, and network diagrams that accompany the textual responses.



Ask your question:

What is the average travel time per line?

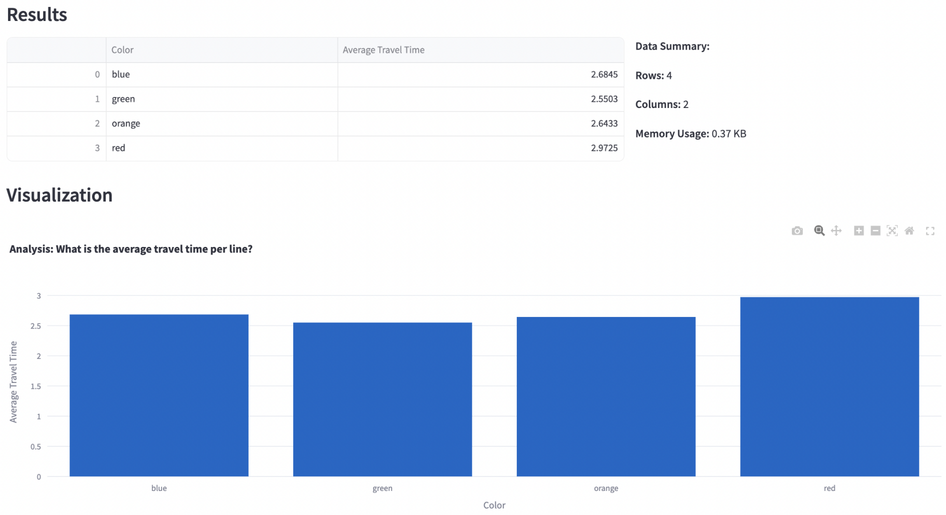
SQL Query Generated!

Generated SQL Query

```
SELECT Color, AVG(Minutes) AS "Average Travel Time" FROM connections GROUP BY Color;
```

Figure 4: Natural language AI Assistant Interaction

This feature not only reduces the technical barrier to data exploration but also enhances decision-making for city planners, transit operators, and commuters. By combining advanced language models with structured transit data, the platform transforms complex datasets into actionable insights accessible to a broad range of users.



Results

	Color	Average Travel Time
0	blue	2.6845
1	green	2.5503
2	orange	2.6433
3	red	2.9725

Data Summary:

Rows: 4

Columns: 2

Memory Usage: 0.37 KB

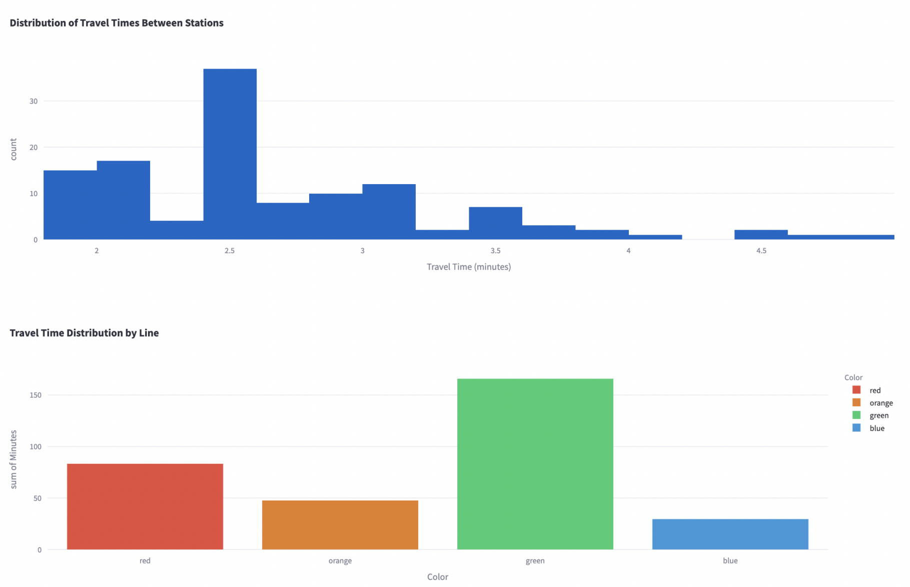
Visualization

Figure 5: Results and Visualization based on AI Assistant’s Response

7 Results

The SmartTransit platform delivered a range of actionable insights into the performance and accessibility of Boston's subway network. By combining GTFS data, supplementary datasets, and AI-powered querying, the system enabled both technical and non-technical users to uncover patterns, diagnose inefficiencies, and identify opportunities for improvement. The following key results were observed:

- **Travel Time Analytics:** Analysis of trip-level data revealed that average peak-hour delays on the Red Line were approximately 15–20% higher compared to off-peak periods. This finding highlights potential areas for schedule optimization and operational adjustments to improve service reliability during high-demand periods.


Distribution of Travel Times Between Stations

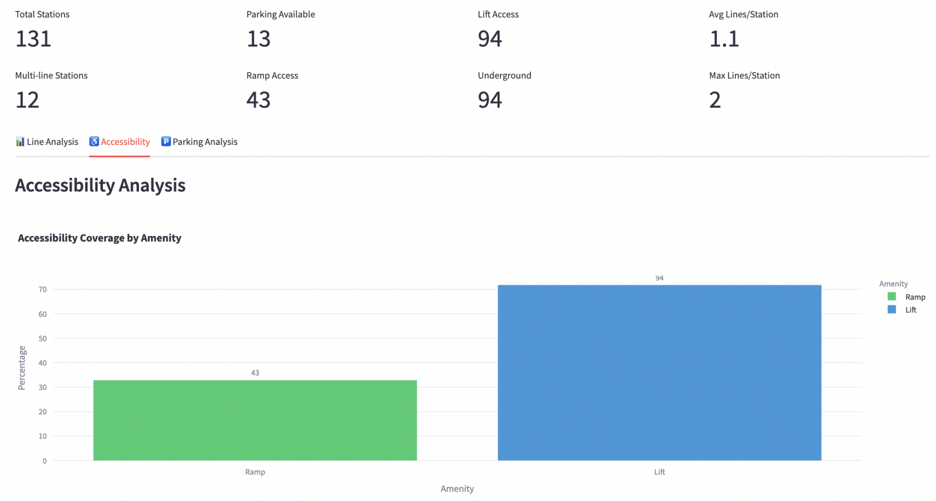
Travel Time Distribution by Line

Figure 6: Travel Time Trend Visualization

- **Accessibility Insights:** The platform mapped station accessibility features, including the presence of elevators, ramps, and wheelchair access points. This dataset supports targeted infrastructure improvements and assists policymakers in prioritizing investments for inclusivity and compliance with accessibility standards.

- **Route Comparison:** Comparative performance metrics were generated for the Red, Green, Orange, and Blue Lines. The analysis revealed that the Blue Line exhibited the most consistent travel times across different periods, suggesting operational practices or infrastructure conditions that may be worth replicating on other lines.

- **AI Query Examples:** The AI-assisted query feature enabled users to ask natural language questions. These queries were automatically translated into SQL using GPT-4, executed against the SQLite database, and returned with relevant visualizations, significantly reducing the need for manual data exploration.



Total Stations
131

Parking Available
13

Lift Access
94

Avg Lines/Station
1.1

Multi-line Stations
12

Ramp Access
43

Underground
94

Max Lines/Station
2

Accessibility Analysis

Figure 7: Accessibility Insights

- **Interactive Visualizations:** The system's Streamlit-powered dashboards provided interactive network graphs, spatial distribution maps, and trend visualizations. These tools empowered non-technical users to explore network density, identify performance bottlenecks, and examine service patterns without requiring advanced technical expertise.

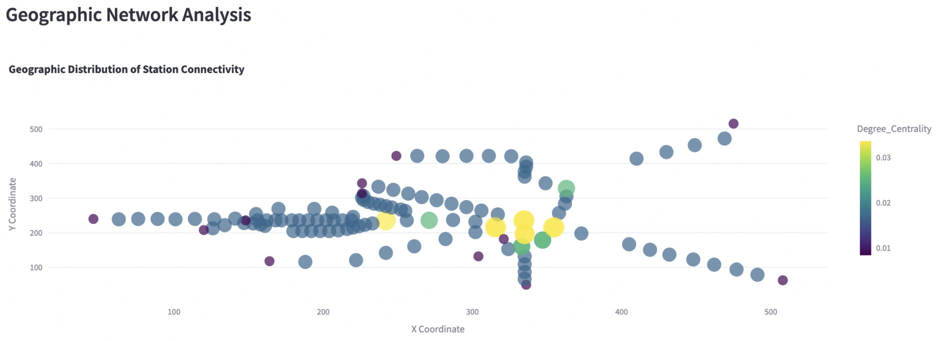
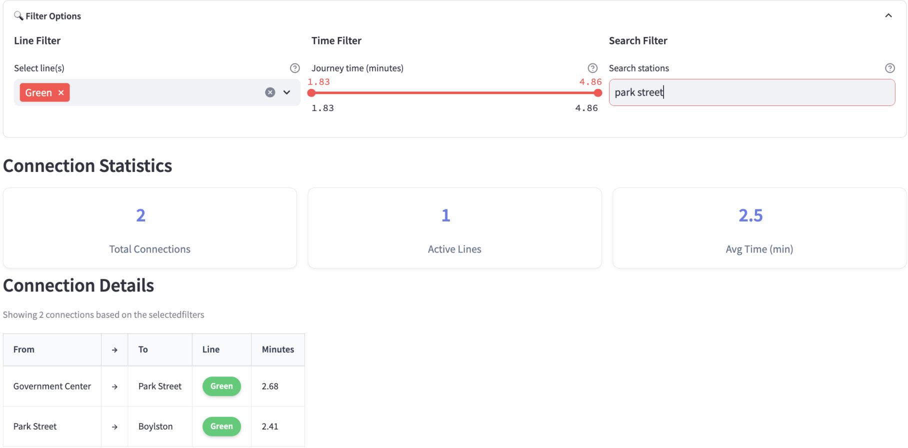


Figure 8: Geographic MBTA Network Analysis Visualization

Collectively, these results demonstrate the platform's capacity to transform static transit datasets into dynamic, interactive resources that inform decision-making for transit planners, operational managers, policymakers, and everyday commuters. The integration of accessibility mapping, comparative performance analytics, and AI-assisted querying represents a significant step toward more data-driven, equitable, and efficient urban transit planning.

8 Conclusion

The SmartTransit project demonstrates the potential of combining open transportation data, graph analytics, and AI-assisted querying to improve the understanding, operation, and planning of urban transit systems. Through its integration of the MBTA's official GTFS feed with supplementary datasets from MassDOT, the platform offers both real-time and historical analyses of subway network performance. These insights are accessible not only to transportation engineers and city planners but also to researchers, policy makers, and commuters seeking to make informed travel decisions. By embedding a natural language query interface powered by AI, the system reduces the technical barrier for data exploration, allowing users to obtain complex analytical results through intuitive interactions. Furthermore, the platform emphasizes accessibility by identifying and highlighting stations equipped with essential facilities such as elevators, ramps, and wheelchair access points, thereby promoting more inclusive transit planning. Overall, SmartTransit provides a scalable and interactive framework that bridges the gap between raw transit data and actionable decision-making.



Filter Options

Line Filter

Select line(s): Green ×

Time Filter

Journey time (minutes): 1.83 – 4.86

Search Filter

Search stations: park street

Connection Statistics

2 Total Connections

1 Active Lines

2.5 Avg Time (min)

Connection Details

Showing 2 connections based on the selectedfilters

From	→	To	Line	Minutes
Government Center	→	Park Street	Green	2.68
Park Street	→	Boylston	Green	2.41

Figure 9: Station Connection Insights from the Interactive Dashboard

9 Limitations

Although the platform achieves its intended objectives, certain limitations constrain its current scope and performance. Firstly, the dataset used is primarily derived from scheduled GTFS feeds and supplementary static datasets, which may not fully reflect real-time conditions such as unexpected delays, maintenance work, or service interruptions. As a result, certain analyses, particularly those related to service reliability, are inherently retrospective rather than predictive. Secondly, the system currently focuses exclusively on MBTA's subway network, limiting its applicability to other modes of public transportation in the region. This means that intermodal connections, such as transfers between subway and bus services, are not fully accounted for in network efficiency analyses. Thirdly, the reliance on in-memory databases, while advantageous for speed, imposes constraints on the volume of data that can be processed simultaneously, which may become a concern as historical datasets grow in size. Additionally, AI-assisted querying, while powerful, is dependent on the accuracy of the underlying natural language processing models, which may occasionally misinterpret complex queries. Finally, accessibility data is reliant on the accuracy and completeness of public datasets, which may not always reflect the current state of station facilities. These limitations provide context for the interpretation of results and guide the directions for future development.

10 Future Work

While the current version of SmartTransit offers a comprehensive analytical toolkit, there remain several avenues for enhancement. One of the immediate priorities is the integration of real-time GTFS feeds, which would enable the platform to provide live updates on service changes, delays, and disruptions, thereby increasing its relevance for day-to-day commuter use. Beyond real-time monitoring, predictive analytics capabilities could be introduced through the development of machine learning models capable of forecasting delays, identifying emerging bottlenecks, and anticipating peak congestion periods. Another important direction is the expansion of the platform to support multi-modal transit analysis, incorporating bus, commuter rail services to present a more holistic view of regional mobility patterns. Additionally, a long-term historical trend analysis module could be implemented to track network evolution, assess the impact of policy changes, and evaluate infrastructure investments over time.

References

1. Google Developers, *General Transit Feed Specification Reference*, 2010. [Online]. Available: `https://developers.google.com/transit/gtfs/reference`.

2. Massachusetts Bay Transportation Authority (MBTA), *MBTA GTFS Data Feeds*, 2025. [Online]. Available: `https://www.mbta.com/developers/gtfs`.

3. Derrible, S. and Kennedy, C., "The Complexity and Robustness of Metro Networks," *Physica A: Statistical Mechanics and its Applications*, vol. 389, no. 17, pp. 3678–3691, 2010. doi: `https://doi.org/10.1016/j.physa.2010.04.008`.

4. Ferreira, N., Poco, J., Vo, H., Freire, J., and Silva, C. T., "Visual Exploration of Big Spatio-Temporal Urban Data: A Study of New York City Taxi Trips," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 12, pp. 2149–2158, 2013. doi: `https://doi.org/10.1109/TVCG.2013.226`.

5. Plotly Technologies Inc., *Plotly Python Graphing Library*, 2025. [Online]. Available: `https://plotly.com/python/`.

6. Streamlit Inc., *Streamlit: The Fastest Way to Build and Share Data Apps*, 2025. [Online]. Available: `https://streamlit.io`.

7. OpenAI, *OpenAI API Documentation*, 2025. [Online]. Available: `https://platform.openai.com/docs/`.

8. Hagberg, A., Schult, D., and Swart, P., *NetworkX: Network Analysis in Python*, 2008. [Online]. Available: `https://networkx.org/`.

GitHUB Repository

https://github.com/MitulNakrani003/DS5110IDMP-Project