Smart City Transit Analytics: Context-Aware Urban Mobility Platform

Vedanth Sathwik Toduru Madabushi & Leo Zheng
CS 4365/6365 - Intro to Enterprise Computing

February 5, 2025

Contents

1 Overview & Motivation

Urban mobility systems face unprecedented challenges in today's rapidly evolving cities. While traditional transit solutions have served well in the past, they struggle to address modern complexities of climate disruptions, event-driven congestion, and evolving commute patterns. Our platform aims to bridge these critical gaps through intelligent, context-aware transit analytics.

1.1 Problem Space

Current urban transit systems face three critical challenges:

1. **Real-time Tracking**: Current MARTA API provides real-time bus/train locations but lacks predictive analytics
2. **Weather Impact**: No integration between weather conditions and transit predictions
3. **User Experience**: Existing apps lack real-time notifications and updates

1.2 Innovation Drivers

Several trends show why we need better solutions:

- **Weather Changes**: More extreme weather events expected in Atlanta by 2030
- **Equal Access**: Some neighborhoods face much longer transit times
- **Economic Impact**: Traffic congestion costs Atlanta $1.8B yearly

1.3 Target Outcomes

Our platform aims to achieve:

- 10% reduction in travel times during busy periods
- Website response time under 5 seconds
- User satisfaction score of 3.5/5 or better

2 Related Work & Technical Foundation

2.1 Existing Solutions Analysis

System	Strength	Limitation	Source
Google Maps	Shows multiple routes	No event information	[Forbes Tech, 2024]
Transit App	Real-time bus tracking	Limited city coverage	[Transit Review, 2023]
MARTA App	Local route information	Basic features only	[MARTA Website]

Table 1: Comparison of Existing Solutions

2.1.1 Limitations of Current Solutions

- **Google Maps**:
  - Lacks real-time MARTA GTFS integration
  - No weather impact considerations
  - Generic routing without local context
- **Transit App**:
  - Missing train tracking capabilities
  - No correlation with weather conditions
  - Limited historical performance analysis
- **MARTA App**:
  - Separate bus and train tracking
  - No predictive capabilities
  - Basic UI without weather overlay

2.1.2 Our Solution's Advantages

- **Unified Tracking**:
  - Combined GTFS feeds for buses and trains
  - Single interface for all transit modes
  - Integrated weather condition display
- **Weather Integration**:
  - Real-time weather impact analysis
  - Historical weather-delay correlations

  - Route recommendations based on conditions

- **Enhanced Predictions**:
  - ML-powered arrival estimates
  - Weather-aware delay predictions
  - Route performance analytics

2.2 Technical Approach

Our solution integrates three key components:

1. **GTFS Processing**:
   - Real-time parsing of MARTA's GTFS feeds
   - Vehicle position normalization and tracking
   - Schedule adherence calculations
2. **Weather Integration**:
   - Open-Meteo API for current and historical weather
   - Impact correlation analysis
   - Route-specific weather sensitivity metrics
3. **Prediction System**:
   - Lightweight regression model for delay predictions
   - Real-time arrival time adjustments
   - Weather-based route recommendations

3 Technical Implementation

3.1 Data Sources

Our implementation leverages available GTFS and weather data:

3.1.1 Transit Data (MARTA GTFS)

- Real-time bus vehicle positions (latitude, longitude, route info)
- Bus trip updates for arrival predictions
- Real-time train tracking data

3.1.2 Weather Data (Open-Meteo)

- Historical weather patterns
- Current weather conditions
- Free API access with no rate limits

3.2 System Architecture

1. **Real-time Integration**:
   - GTFS feed parsing and normalization
   - WebSocket updates for vehicle positions
   - Weather condition monitoring
2. **Data Pipeline**:
   - Vehicle position tracking and updates
   - Weather impact correlation analysis
   - Delay prediction based on weather patterns

   Data Models:

```
interface Bus {
    adherence: string;
    direction: string;
    latitude: string;
    longitude: string;
    route: number;
    stop_id: string;
    timepoint: string;
    trip_id: string;
    vehicle: string;
}

```

```
interface Train {
    destination: string;
    direction: string;
    line: string;
    next_arrival: Date;
    station: string;
    train_id: string;
    waiting_seconds: string;
    waiting_time: string;
}
```

3.3 Machine Learning Pipeline

Our model focuses on correlating weather patterns with transit delays:

3.4 Machine Learning Pipeline

Our predictive model focuses on establishing correlations between weather conditions and transit system performance. By analyzing historical patterns and real-time data, we aim to provide accurate predictions for transit delays under various weather conditions.

3.4.1 Available Features

- **Historical Weather Metrics**: Temperature variations, precipitation levels, and historical weather patterns from Open-Meteo's database to establish baseline weather impacts
- **Vehicle Adherence**: GTFS data showing how well buses and trains maintain their schedules, including deviation from planned arrival times
- **Real-time Position Updates**: Continuous GPS tracking data from MARTA's GTFS feed, providing current vehicle locations and movement patterns
- **Route-specific Patterns**: Historical performance data for individual routes, identifying segments particularly susceptible to weather-related delays
- **Time-based Factors**: Peak hours, day of week, and seasonal patterns that may influence weather's impact on transit

3.4.2 Prediction Goals

- **Weather-based Delay Probabilities**: Calculate the likelihood of delays based on current and forecasted weather conditions, providing confidence intervals for predictions
- **Route Performance Analysis**: Evaluate how specific routes respond to different weather conditions, identifying weather-sensitive segments
- **Real-time Arrival Predictions**: Combine weather data with current vehicle positions to provide more accurate arrival time estimates

- **Early Warning System**: Alert users about potential weather-related delays before they occur, enabling better trip planning

3.4.3 Implementation Approach

We will implement this pipeline using a lightweight regression model trained on recent historical data. The model will be retrained weekly to adapt to changing patterns, with daily updates incorporating the latest weather and transit performance data. This approach balances prediction accuracy with the computational constraints of our 6-week development timeline.

4 Development Plan

4.1 Implementation Schedule

Week	Tasks	Owner	Deliverable
1	GTFS Integration	Both	Data pipeline
2	Weather API	Leo	Historical analysis
3	Real-time Map	Sathwik	Live tracking
4	ML Training	Leo	Delay predictions
5	UI/UX Polish	Sathwik	Mobile design
6	Deployment	Both	Production launch

Table 2: 6-Week Project Timeline

4.2 Technology Stack

Component	Technology	Purpose
Frontend	React	Real-time mapping
Backend	Node.js + WebSocket	Live updates
APIs	MARTA + Open Meteo	Transit & weather data
Database	CSV	Easy storage and retrieval
Deployment	Vercel	Hosting

Table 3: Revised Technology Stack

5 Skill Development & Team Structure

5.1 Learning Objectives

5.1.1 Transit Data Skills

- **GTFS Protocol**:
  - Parsing real-time GTFS feeds
  - Protocol buffer implementation
  - Trip and route data structures
- **Geospatial Processing**:
  - Coordinate system handling
  - Real-time position tracking
  - Map visualization techniques

5.1.2 API Integration

- **Weather API**:
  - Open-Meteo data retrieval
  - Historical data analysis
  - Weather condition mapping
- **Real-time Data**:
  - WebSocket implementations
  - Stream processing
  - Data synchronization

5.1.3 Machine Learning

- **Time Series Analysis**:
  - Weather pattern correlation
  - Transit delay prediction
  - Model optimization

5.2 Team-Specific Development

5.2.1 Sathwik's Focus Areas

- Real-time map visualization

- GTFS feed processing
- Frontend state management
- Learning Resources:
  - GTFS documentation
  - React state patterns

5.2.2 Leo's Focus Areas

- Weather API integration
- ML model development
- Data pipeline architecture
- Learning Resources:
  - Open-Meteo API docs
  - Time series analysis
  - Stream processing patterns

5.3 Collaborative Approach

- Daily GTFS data reviews
- Weather impact analysis sessions
- Code review focusing on real-time features
- Documentation of API integration patterns

6 Deliverables & Validation

6.1 Deliverables

1. **Transit Tracking System**:
    - Real-time vehicle position map
    - Weather condition overlay
    - Delay predictions dashboard
2. **Analysis Tools**:
    - Route performance metrics
    - Weather impact visualizations
    - Prediction accuracy reports
3. **Documentation**:
    - API integration guides
    - ML model documentation
    - System architecture diagrams

6.2 Success Metrics

Metric	**Target**	**Method**
Update Frequency	< 30s	WebSocket logs
Prediction Accuracy	RMSE < 3min	Actual vs Predicted
Data Freshness	< 60s	API monitoring
System Uptime	99%	Vercel analytics

Table 4: Performance Metrics

7 Ethical Considerations

7.1 Privacy and Security

1. **Data Protection**:
    - No personal user data collection
    - Public transit data only
    - Simple password protection for admin access
2. **Transparency**:
    - Open source code on GitHub
    - Clear documentation of data sources
    - Simple explanation of how predictions work

7.2 Community Impact

- **Accessibility**: Mobile-friendly website design
- **Language Support**: Basic Spanish translation
- **User Privacy**: No user tracking or cookies

8 Works Cited

References

[1] Metropolitan Atlanta Rapid Transit Authority. (2024). *Service Changes Effective December 2024*. https://www.itsmarta.com/servicechanges.aspx

[2] National Weather Service. (2023). *Summer 2023 Climate Summary for Metropolitan Atlanta*. NOAA Technical Report NWS-SR-2023-15. https://www.weather.gov/ffc/Summer2023ClimateSummary

[3] Atlanta Regional Commission. (2024). *Comprehensive Economic Development Strategy: 2024 Annual Update*. https://cdn.atlantaregional.org/wp-content/uploads/ceds-annual-update-draft-4.pdf

[4] City of Atlanta. (2023). *Climate Action Plan: Roadmap to Resilience*. https://atlantaclimateactionplan.wordpress.com/wp-content/uploads/2016/02/atlanta-climate-action-plan-07-23-2015.pdf

[5] Vizologi. (2023). *Transit App Business Model Analysis*. https://vizologi.com/business-strategy-canvas/transit-app-business-model-canvas/

[6] IEEE Standards Association. (2023). *Ethically Aligned Design: A Vision for Prioritizing Human Well-being with Autonomous and Intelligent Systems*. IEEE Std 7000-2023.

[7] National Institute of Standards and Technology. (2024). *Artificial Intelligence Risk Management Framework*. NIST AI 100-1.

[8] Zhao, J., & Chen, L. (2023). *Reinforcement Learning for Adaptive Transit Systems*. MIT Press.

[9] Transportation Research Board. (2024). *Hybrid Machine Learning Applications in Public Transit*. TRB Special Report 345.

[10] Obe, R.O., & Hsu, L.S. (2023). *PostGIS in Action: Geographic Object-Relational Modeling*. Manning Publications.

[11] Forbes Tech Council. (2019). *Intelligent Transportation Technology and Quality of Life Improvements*. https://www.forbes.com/councils/forbestechcouncil/2019/04/03/intelligent-transportation-technology-and-your-quality-of-life/

[12] Atlanta Transit Equity Coalition. (2024). *Regional Mobility Justice Report*. https://www.letspropelatl.org/atlanta-transit-equity-coalition

[13] National Renewable Energy Laboratory. (2023). *Mobility Energy Productivity Metric Handbook*. NREL/TP-5400-80903.

[14] National Weather Service. (2023). *Annual Climate Summary for Metropolitan Atlanta*. NOAA Technical Report NWS-SR-2023-20. https://www.weather.gov/ffc/2023AnnualClimateSummary

[15] Metropolitan Atlanta Rapid Transit Authority. (2024). *MARTA Developer Resources - Real-Time APIs*. https://www.itsmarta.com/app-developer-resources.aspx

[16] Open-Meteo. (2024). *Weather Forecast API Documentation*. https://open-meteo.com/en/docs