Railway Network Intelligence: An AI-Driven Approach to Network Analysis and Optimization

Umar Aslam

December 29, 2024

Abstract

This report presents a comprehensive analysis of an AI-powered railway network intelligence system. The system combines graph theory, machine learning, and real-time data analytics to monitor, analyze, and optimize railway network operations. We implement multiple ML models including Isolation Forest for anomaly detection, DBSCAN for track clustering, and linear regression for delay prediction. The system provides real-time insights through an interactive dashboard, enabling efficient network management and predictive maintenance scheduling.

Contents

1 Introduction

Railway networks are complex systems requiring continuous monitoring and optimization. This project implements an intelligent system that:

- Detects anomalous track behavior using unsupervised learning
- Clusters similar tracks based on operational characteristics
- Predicts future delays using time series analysis
- Visualizes network health through an interactive dashboard

2 Mathematical Models

2.1 Network Representation

The railway network is modeled as a graph $G = (V, E)$ where:

- V represents the set of stations
- E represents the set of tracks connecting stations

Each edge $e_{ij} \in E$ has attributes:

$$e_{ij} = \{c_{ij}, s_{ij}, m_{ij}, \eta_{ij}, h_{ij}\} \tag{1}$$

where:

- c_{ij} is the capacity
- s_{ij} is the speed limit
- m_{ij} is the maintenance score
- η_{ij} is the energy efficiency
- h_{ij} is the historical delay vector

2.2 Anomaly Detection

We implement Isolation Forest for anomaly detection. For each track, we create a feature vector:

$$\mathbf{x}_{ij} = [c_{ij}, \bar{h}_{ij}, m_{ij}, \eta_{ij}]^T \tag{2}$$

The anomaly score is calculated as:

$$s(\mathbf{x}, n) = 2^{-\frac{E(h(\mathbf{x}))}{c(n)}} \tag{3}$$

where:

- $h(\mathbf{x})$ is the path length
- $c(n)$ is the average path length
- n is the number of samples

2.3 Track Clustering

DBSCAN clustering is applied using normalized features:

$$\mathbf{f}_{ij} = [\frac{s_{ij}}{s_{max}}, m_{ij}, \eta_{ij}]^T \tag{4}$$

The clustering criteria are:

$$N_\epsilon(\mathbf{p}) = \{\mathbf{q} \in D | \text{dist}(\mathbf{p}, \mathbf{q}) \leq \epsilon\} \tag{5}$$

2.4 Delay Prediction

Linear regression is used for delay prediction:

$$\hat{y} = \mathbf{w}^T \mathbf{x} + b \tag{6}$$

where:

- $\mathbf{x}$ is the time point vector
- $\mathbf{w}$ are the learned weights
- b is the bias term

3 Implementation

3.1 Network Generation

The network is generated using the Watts-Strogatz model with parameters:

- N: number of nodes
- K: mean degree
- β: rewiring probability

Algorithm 1 Network Generation

```
Initialize Watts-Strogatz graph G(N, K, β)
for each edge (i, j) in G do
    Assign capacity c_ij ~ U(50, 100)
    Assign speed limit s_ij ~ U(60, 300)
    Assign maintenance score m_ij ~ U(0.6, 1.0)
    Generate historical delays h_ij
end for
```

3.2 Anomaly Detection Pipeline

The anomaly detection process involves:

1. Feature extraction from track attributes
2. Feature normalization using StandardScaler
3. Isolation Forest model fitting
4. Anomaly score calculation

Algorithm 2 Anomaly Detection

1: Extract features $\mathbf{X}$
2: $\mathbf{X}_{norm}$ = StandardScaler($\mathbf{X}$)
3: Initialize IsolationForest(contamination=0.1)
4: Fit model and predict anomalies
5: Return anomaly labels

3.3 Track Clustering Process

Track clustering follows these steps:

1. Feature normalization
2. DBSCAN clustering ($\epsilon = 0.3$, min_samples=2)
3. Cluster assignment

4 Results and Analysis

4.1 Network Statistics

Key metrics from the implementation:

- Average node degree: 3
- Network density: varies with size
- Clustering coefficient: approximately 0.3

4.2 Anomaly Detection Performance

The Isolation Forest model typically identifies:

- 10% of tracks as anomalous
- Focuses on tracks with unusual combinations of:
  - High delays with good maintenance
  - Low efficiency with high capacity
  - Inconsistent performance metrics

4.3 Clustering Results

DBSCAN typically identifies:

- 2-4 main track clusters
- Noise points (unique tracks)
- Clear separation between high-speed and freight tracks

5 Visualization Dashboard

5.1 Components

The dashboard consists of four main components:

1. Network Health Monitor
2. Track Classification Map
3. Performance Radar Chart
4. Delay Forecast Plot

5.2 Interactive Features

Key interactive elements include:

- Hover information for tracks and stations
- Real-time updates of metrics
- Color-coded status indicators
- Predictive analytics display

6 Future Work

Potential improvements include:

- Implementation of deep learning models
- Real-time data integration
- Advanced optimization algorithms
- Enhanced visualization features

7 Conclusion

This project demonstrates the effectiveness of combining machine learning with network analysis for railway system optimization. The implemented system provides valuable insights for network management and maintenance planning.

A Implementation Details

A.1 Code Structure

Key classes and methods:

```
RailwayNetworkSimulator
 _create_network()
 _generate_train_data()
 detect_anomalies()
 cluster_similar_tracks()
 predict_future_traffic()
 visualize_network()
```