

BCSE497J Project-I

Railway Rescheduling Automation System

22BCE0984

PREETI AMBWANI

22BDS0009

ADITYA SHARMA

Under the Supervision of

Professor Tamizharasi T

Designation

School of Computer Science and Engineering (SCOPE)

B.Tech.

in

Computer Science and Engineering

(with specialization in CSE Core and Data Science)

School of Computer Science and Engineering



September 2025

ABSTRACT

The Railway Rescheduling Automation System, or RRAS, is a data-driven system that is meant to minimize train delays and maximize the efficiency of railway operations. It integrates a number of APIs to source information about passenger volumes, current weather conditions, weather forecasts for the future, scheduled and actual arrival and departure times, and the duration of stays at stations. By observing how weather conditions correspond to delay in operations, the system generates predictive models that can detect potential problems in future schedules. Based on these forecasts, RRAS builds better schedules by verifying the aggregate delays at each station every 15 minutes for a four-hour interval. The best rescheduling plan is then selected and displayed as an outline timetable for the next day. The system is developed with Python, Flask, NumPy, Pandas, and Scikit-Learn to provide a smooth way of handling data and sophisticated predictive analysis. Through this project, it can be illustrated that the application of machine learning with real-time data helps enhance pre-scheduling of trains, reduce the inconvenience caused to commuters, and optimize railway operations.

TABLE OF CONTENTS

Sl.No	Contents	Page No.
	Abstract	2
1.	INTRODUCTION	4
	1.1 Background	4
	1.2 Motivations	4
	1.3 Scope of the Project	4
2.	PROJECT DESCRIPTION AND GOALS	5
	2.1 Literature Review	5
	2.2 Gaps Identified	5
	2.3 Objectives	6
	2.4 Problem Statement	6
	2.5 Project Plan	6
3.	REQUIREMENT ANALYSIS *	8
	3.1 Functional Requirements	8
	3.2 Non-Functional Requirements	9
	3.3 Hardware & Software Specifications	9
4.	SYSTEM DESIGN*	10
	4.1 Workflow Model	10
	4.2 Module Design and Implementation	11
5.	REFERENCES	11

Note *

Respective guides can decide on the contents to be prepared in consultation with the students.

INTRODUCTION

1.1 Background

Railways are one of the most widely used modes of transport across the world, carrying millions of passengers daily. However, delays in train schedules due to unpredictable weather conditions, overcrowding, or operational inefficiencies often cause significant inconvenience to passengers. Traditional railway scheduling systems are mostly static and do not incorporate real-time factors such as crowd density and weather predictions. As a result, delays propagate across the network, leading to operational inefficiencies and poor passenger satisfaction.

1.2 Motivations

The motivation behind this project lies in addressing one of the most common issues faced by railway systems: delays. With the increasing availability of real-time data through APIs and the advancement of machine learning models, it has become feasible to predict and mitigate delays before they occur. By incorporating weather forecasts, crowd levels, and operational records into predictive algorithms, train scheduling can be made adaptive and intelligent. This will not only reduce uncertainty for passengers but also help railway operators improve overall efficiency.

1.3 Scope of the Project

The scope of RRAS is limited to prediction and rescheduling. It collects and integrates relevant data from multiple sources, analyzes the association between delays and external conditions, and generates optimized timetables for the following day. The project does not include real-time signaling or physical train control, but rather focuses on decision support for railway operators.

PROJECT DESCRIPTION AND GOALS

2.1 Literature Review

Recent research in train delay prediction highlights the role of machine learning and spatio-temporal models. For example, Oneto et al. (2016) demonstrated that including weather data significantly improves prediction accuracy. Similarly, Zhang et al. (2024) proposed a Bayesian Spatio-Temporal Graph Convolutional Network for predicting railway delays, showing strong results in handling delay propagation. Other works, such as those using ensemble methods and Markov chains, provide insights into the statistical and machine learning approaches used in delay modeling. Despite these advancements, few systems integrate weather, crowd levels, and dwell time simultaneously, which creates an opportunity for RRAS.

2.2 Gaps Identified

- Many existing systems concentrate solely on past delay information without taking into account environmental or operational conditions.
- Not many research efforts combine information about crowd density with forecasts of delays.
- Practical rescheduling methods are restricted, as most of them focus only on predicting delays rather than taking steps to fix them.

Existing Gap:

Most train scheduling and delay prediction studies focus on timetables, historical delays, and sometimes weather data. However, a crucial challenge in real-world implementation is the unavailability of accurate latitude and longitude data for railway stations. APIs often provide station codes or textual names, but geospatial coordinates are either missing or inconsistent across sources. This hinders:

- Mapping weather data (which is usually location-based) to specific stations.
- Using spatial models such as Graph Neural Networks (GNNs) for delay propagation.
- Visualizing routes on a map or integrating with GIS tools.

How RRAS Addresses It:

Our system incorporates an additional station geocoding module that queries open datasets or third-party geocoding APIs (e.g., Google Maps API, OpenStreetMap Nominatim) to fetch and standardize station coordinates. This step ensures that weather forecasts, crowd data, and

operational delays are correctly linked to the station's geographical position, enabling more accurate delay prediction and rescheduling.

2.3 Objectives

- To collect and integrate operational and environmental data (crowd levels, weather, timetables, dwell times).
- To establish correlations between weather and train delays.
- To use predictive modeling to forecast delays for the next day.
- To generate optimized train schedules that minimize cumulative delays.
- To provide a decision support system for railway operators.

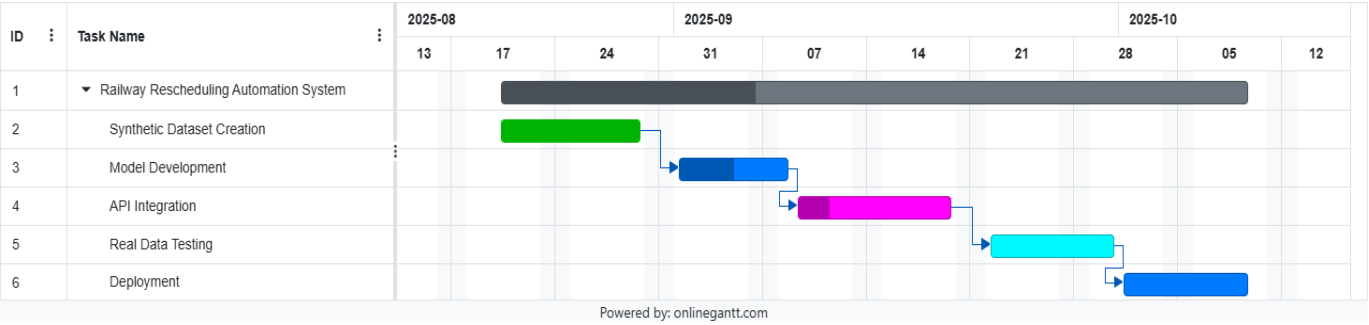
2.4 Problem Statement

Train delays are a major challenge for railway networks, leading to inefficiencies and poor passenger satisfaction. Current systems lack predictive adjustments that account for environmental and operational factors. There is a need for a data-driven framework that not only predicts delays but also provides optimized schedules in advance.

2.5 Project Plan

- Phase 1: Synthetic Dataset – Create a synthetic dataset with station details, weather, and delays to test model feasibility.
- Phase 2: Model Development – Train and validate delay prediction and rescheduling algorithms on synthetic data.
- Phase 3: API Integration – Incorporate real-world APIs for railway schedules, weather forecasts, crowd levels, and station geocoding.
- Phase 4: Real Data Testing – Evaluate performance with live data for accuracy and scalability.
- Phase 5: Deployment – Provide a Flask-based web interface to present optimized schedules.

Gantt Chart:



The Gantt chart shows the Railway Rescheduling Automation System timeline from mid-August to early October 2025, progressing sequentially from dataset creation to model development, API integration, real data testing, and final deployment, with each phase dependent on the previous one.

REQUIREMENT ANALYSIS

3.1 Functional Requirements

ID	Requirement	Priority	Category	Justification
FR-1	Collect train schedules and operational data.	High	Data Input	Provides essential data for analysis and predictions.
FR-2	Gather weather and passenger density data.	High	Data Input	External factors like weather and crowding impact train delays.
FR-3	Use machine learning to predict potential delays	High	Processing	Enables proactive decisions by forecasting disruptions.
FR-4	Generate optimized timetables to minimize delays.	High	Processing	Ensures smoother operations and reduces passenger inconvenience.
FR-5	Provide a web interface to display schedules.	Medium	User Output	Easy access for operators and stakeholders.

3.2 Non-Functional Requirements

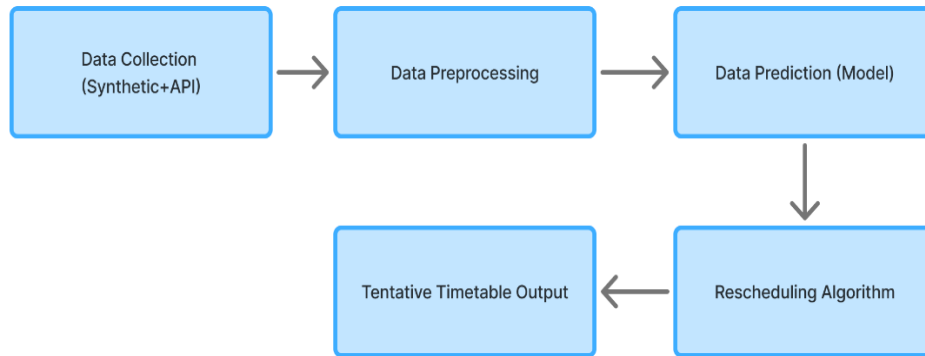
ID	Requirement	Priority	Category	Justification
NFR-1	System must handle multiple trains and stations.	High	Performance	Scalable for future network expansion.
NFR-2	Predictions should be accurate and consistent.	High	Reliability	Ensures trust and usability of the system.
NFR-3	Interface must be intuitive for users.	Medium	Usability	Reduces training time and improves adoption.
NFR-4	Minimize computational overhead.	High	Efficiency	Supports timely results for operational decisions.

3.3 Hardware & Software Specifications

- Language: Python
- Framework: Flask
- Libraries: NumPy, Pandas, Scikit-Learn
- APIs: Weather APIs, Railway APIs
- Hardware: Laptop/PC with minimum Intel i5, 8GB RAM

SYSTEM DESIGN

4.1 Workflow Model



Explanation of Workflow Diagram

1. **Data Collection (Synthetic + API)**

The process starts with gathering information such as train schedules, operational details, weather conditions, and passenger density. Both synthetic datasets and live API feeds are used to ensure the system works in training as well as in real-time.

2. **Data Preprocessing**

The collected data is cleaned and organized to remove errors or gaps. Useful features are extracted so the data is ready for accurate model predictions.

3. **Data Prediction (Model)**

Machine learning models then analyze the processed data to predict possible delays. These predictions act as the basis for smarter scheduling decisions.

4. **Rescheduling Algorithm**

Using the predictions, the system applies an optimization algorithm to create updated timetables that minimize delays while keeping train operations practical.

5. **Tentative Timetable Output**

Finally, the system generates a clear and accessible timetable that can be viewed by operators through a user-friendly interface. This allows railway staff to review and act on the updated schedules.

4.2 Module Design and Implementation

- **Data Collection Module:** Synthetic dataset + API integration.
- **Preprocessing Module:** Clean and prepare data for modeling.
- **Prediction Module:** ML-based delay forecasting.
- **Rescheduling Module:** Optimize schedules over 15-min intervals.

(Currently ~40% completed with synthetic data + prediction model implementation).

REFERENCES

1. Oneto, L., et al. “**Advanced Analytics for Train Delay Prediction Systems by Including Exogenous Weather Data.**” *IEEE DSAA*, 2016.
2. Zhang, J., et al. “**Bayesian Spatio-Temporal Graph Convolutional Network for Railway Train Delay Prediction.**” *IEEE Transactions on Intelligent Transportation Systems*, 2024.
3. Sun, Z., et al. “**Train Time Delay Prediction for High-Speed Train Dispatching Based on Spatio-Temporal Graph Convolutional Network.**” *IEEE T-ITS*, 2022.
4. Ambrosio, A., et al. “**A Comparative Analysis of Train Delay Prediction Models for Markov Chains.**” *Transportation Research Procedia*, 2025.
5. Li, J., et al. “**Real-Time Adjustment Method for Metro Systems with Train Delays Based on Improved Q-Learning.**” *Applied Sciences*, MDPI, 2024.