PROJECT ABSTRACT

Bus Arrival Time Prediction using GPS History

Project ID: k-1642 | projectech.in/projects/bus-arrival-time-prediction/

Commuters lose hours to unpredictable buses: static timetables ignore traffic, dwell times and bunching. This project learns arrival times from historical GPS traces using gradient-boosted regression on segment-level features, and presents live predictions on an arrival board and tracking map with reliability statistics per route.

Suitable for B.E./B.Tech final-year projects in Computer Science, IT and AI & Machine Learning.

Project type: Software

Availability: Built to order

Suitable branches: Computer Science, IT and AI & Machine Learning

Technologies: Python 3, scikit-learn, Gradient Boosting, GTFS/GTFS-Realtime, pandas, OpenStreetMap, HTML/JS live board

Price: Request a quotation

Prepared by Projectech · projectech.in — for academic project reference.

1. Title

Bus Arrival Time Prediction using GPS History — a built-to-order software project delivered with a working demo application, complete source code, project report, presentation and viva preparation kit.

2. Introduction / Background

Published bus schedules are static, but travel times are not. Riders cannot tell whether a bus is 2 or 20 minutes away, so they wait at stops or over-allocate buffer time. Transit agencies log GPS probes (GTFS-Realtime), but turning raw traces into accurate, explainable arrival predictions — handling dwell time, segment speeds and time-of-day effects — is a real data-science pipeline students can build end to end.

3. Problem Statement

Published bus schedules are static, but travel times are not. Riders cannot tell whether a bus is 2 or 20 minutes away, so they wait at stops or over-allocate buffer time. Transit agencies log GPS probes (GTFS-Realtime), but turning raw traces into accurate, explainable arrival predictions — handling dwell time, segment speeds and time-of-day effects — is a real data-science pipeline students can build end to end.

4. Objectives

- Ingest historical bus GPS traces and map them to route segments and stops.

- Engineer segment features: historical speed, time-of-day, day-of-week, dwell time, headway.
- Train a gradient-boosting regressor for segment travel time and stop arrival.
- Evaluate MAE against the schedule baseline on a held-out period.
- Deliver a live arrival board, tracking map and per-route reliability view.

5. Proposed Methodology

GPS probes are map-matched to route segments; each segment traversal becomes a training row with engineered features. A gradient-boosting regressor (or comparable ensemble) is trained per route group with time-based cross-validation to avoid leakage. Arrival at a downstream stop is the sum of predicted segment times plus dwell estimates. The held-out evaluation reports MAE overall and by peak/off-peak, compared with the static-schedule baseline.

6. System Architecture / Working

A Python pipeline (ingestion, map-matching, feature building, training, evaluation) feeds a single-file HTML/JS app: the arrivals tab lists live predicted arrivals per stop with confidence; the map tab animates buses along routes; the reliability tab shows on-time percentages and MAE by route and hour. All served from static files after the model artifacts are generated.

7. Hardware / Software Requirements

- Hardware: any machine that runs Python 3 (4 GB RAM minimum).
- Software: Python 3, scikit-learn, pandas, NumPy.
- Data: historical GPS traces (GTFS-Realtime archive or simulated traces documented in the report).
- Demo app: any modern web browser; runs offline after download.

8. Expected Outcomes

- A trained arrival-time model with a design-target MAE under 90 seconds on held-out data.
- A live arrival board and tracking map that make predictions tangible.
- A reliability report quantifying improvement over the static timetable.
- A documented map-matching and feature pipeline reusable for other transit analytics.

9. Applications

- Academic final-year project in Computer Science, IT or AI & Machine Learning programs.
- Teaching aid for applied regression, time-based validation and geospatial data.
- Prototype for a college or city bus tracking display.

10. Future Scope

- Real-time GTFS-Realtime feed integration.
- Passenger-load prediction from ticketing data.
- Bus-bunching detection and headway alerts.
- Multimodal trip planning with predicted transfers.