JETNR
Journal of Emerging Trends and Novel Research
JETNR.ORG | ISSN : 2984-9276
An International Open Access, Peer-reviewed, Refereed Journal

LIVE KSRTC BUS TRACKER WITH SMART ETA PREDICTION: A SIMULATION-BASED APPROACH TO INTELLIGENT PUBLIC TRANSPORT SYSTEMS

[1]**Simran M**, [2]**Mohammed Aafaq**, [3]**Taniya Khanum**, [4]**Md Imtiyaz**
[5]**Mrs. Megharani H M**, [5]Assistant Professor and project guide,
[12345] Department of Computer Science and Engineering,
Sri Siddhartha School of Engineering, Tumakuru, Karnataka, India
[1]simranm3116@gmail.com, [2]mohammedaafaq719@gmail.com, [3] taniyakhanum510@gmail.com, [4] moadimtiyaz@gmail.com

Abstract : The Karnataka State Road Transport Corporation (KSRTC) operates one of the largest public bus networks in India, serving millions of passengers across Karnataka daily. Despite this scale, passengers are deprived of real-time bus location information, resulting in prolonged and uncertain waiting times that erode trust in public transportation. Deploying a production-grade tracking system requires GPS-enabled hardware across the entire fleet, an investment beyond the feasibility of an academic prototype. This paper proposes the design and development of a Simulated Real-Time KSRTC Bus Tracking and ETA Prediction System that demonstrates the working principles of Intelligent Transport Systems (ITS) without requiring physical GPS devices. The system employs a software-based GPS simulation module emulating live bus movement along predefined KSRTC routes. Simulated location data is transmitted to a backend server via WebSocket, processed using map-matching and the Haversine distance formula, and broadcast to a web-based interface built with Leaflet.js. Estimated Time of Arrival (ETA) is computed as remaining route distance divided by current average speed. The architecture is modular — the GPS simulator is replaceable by real GPS hardware in future deployments with zero changes to the backend or frontend. Preliminary evaluation shows ETA mean absolute error of 11.4 seconds at constant speed and WebSocket update latency under 80 ms for 10 concurrent clients, demonstrating adequate accuracy and performance for a pilot deployment.

Index Terms — Real-time bus tracking, GPS simulation, ETA prediction, KSRTC, Haversine formula, WebSocket, Leaflet.js, Intelligent Transport Systems, public transportation, Karnataka.

I. INTRODUCTION

Public transportation systems are the backbone of urban mobility in developing nations. In India, state-operated bus networks such as the Karnataka State Road Transport Corporation (KSRTC) provide an affordable, accessible alternative to private vehicles, serving thousands of routes and millions of passengers daily. Despite the critical role of these networks, passengers frequently experience uncertainty regarding bus arrival times due to the complete absence of real-time tracking information at the passenger level.

The impact of this information gap is significant. Studies on public transit behaviour consistently demonstrate that uncertainty about wait times is perceived as more stressful than actual waiting time [1]. Passengers who cannot predict bus arrivals either arrive excessively early, wasting time, or miss buses altogether due to last-minute uncertainty. This poor experience directly contributes to declining ridership and increased dependence on private vehicles, worsening urban traffic congestion and carbon emissions.

Intelligent Transport Systems (ITS) offer a technological solution to this challenge. Modern ITS deployments leverage GPS technology to broadcast real-time vehicle locations to centralised servers, which compute Estimated Time of Arrival (ETA) and disseminate this information to passengers via mobile and web applications. However, full-scale GPS deployment across an entire bus fleet requires significant hardware investment, infrastructure coordination, and regulatory approvals — constraints that make such systems impractical for academic research prototypes.

This paper addresses the gap between the demonstrated feasibility of real-time bus tracking in research literature and the absence of a working, simulation-based prototype for the KSRTC network. Our system replaces physical GPS hardware with a software-based GPS simulation module that generates realistic location data streams for buses moving along predefined KSRTC routes. All other components — backend server, WebSocket broadcasting, Haversine ETA computation, and Leaflet.js web interface — function identically to a production deployment. The simulator is a drop-in module, replaceable by real GPS hardware with zero changes to the rest of the system.

The remainder of this paper is organised as follows: Section II reviews related literature; Section III describes the system architecture and methodology; Section IV details the implementation; Section V presents results and discussion; Section VI concludes with future work directions.

II. LITERATURE REVIEW

A substantial body of research has explored real-time bus tracking and ETA prediction using various hardware and algorithmic approaches. A thorough review of six key papers reveals consistent patterns in methodology as well as critical gaps that the proposed system addresses.

Upendra Babu et al. (2022) [2] presented a real-time bus management system using Raspberry Pi, GPS, and GSM modules. The system employed Support Vector Regression (SVR) to predict ETA, incorporating traffic density and weather as input features, achieving an RMSE of approximately 27 seconds. While demonstrating the viability of ML-based ETA prediction for Indian transit, it relies entirely on embedded GPS hardware installed in each bus, making academic replication impractical.

Jayaraam et al. (2024) [3] proposed a Smart Bus Route Management System integrating Kalman Filter-based GPS noise reduction, Hidden Markov Model (HMM) map-matching, and Gradient Boosting algorithms (XGBoost and LightGBM) for ETA prediction. This paper presents the most technically comprehensive pipeline reviewed, covering the full data flow from raw GPS input to user-facing predictions. However, it was designed for generic urban deployments without KSRTC-specific route data or Karnataka network constraints.

Vidanapathirana et al. (2023) [4] validated a GPS-based data collection system for public transit using hardware and mobile applications. Critically, this work explicitly acknowledges the validity of simulation-based data collection as an interim approach in the absence of deployed hardware — directly supporting the methodological choice in this paper.

A further IRJMETS paper (2024) [3] described a web-based real-time tracking system integrating GPS, IoT sensors, and the Google Maps Distance Matrix API for ETA computation. The system uses WebSocket communication via Laravel Echo and Leaflet.js for rendering — an architecture remarkably close to the one proposed here. The primary gap is the absence of a simulation module for hardware-free prototyping.

A foundational study by Marques-Neto et al. (2007) [5] presented one of the earliest GPS-based ETA estimation algorithms, implemented in the SITCUO urban bus transit system in Brasilia. The methodology uses speed and distance from GPS data to estimate arrival times, forming the conceptual basis for the Haversine-based ETA formula used in this work.

Nesmachnow and Gomez (2023) [6] proposed a methodology for generating synthesised GPS data for public transportation systems using Generative Adversarial Networks (GANs). This work validates the use of synthetic, simulation-generated location data as a legitimate input for transit research, reinforcing the academic credibility of our GPS simulation approach.

Table I: Summary of Related Literature

Ref.	Authors & Year	Method / Technology	Key Gap
[2]	Upendra Babu et al., 2022	SVR + GPS/GSM (Raspberry Pi)	Requires physical GPS hardware
[3]	Jayaraam et al., 2024	Kalman Filter + HMM + XGBoost	No KSRTC-specific route data
[4]	Vidanapathirana et al., 2023	GPS hardware + mobile app	Hardware dependent; no simulation
[3]	Jayaraam S.P. & Jayakumar, 2024	WebSocket + Leaflet.js + API	No simulator; requires bus GPS
[5]	Marques-Neto et al., 2007	Speed/distance ETA (SITCUO)	Requires fleet-wide GPS hardware
[6]	Nesmachnow & Gomez, 2023	GAN synthetic transit data	Generic; no KSRTC modelling

The literature review confirms that no existing system provides a simulation-based, hardware-free, KSRTC-specific bus tracking and ETA prediction prototype. This constitutes the primary research gap addressed in this paper.

III. SYSTEM ARCHITECTURE AND METHODOLOGY

The proposed system comprises five interconnected components forming an end-to-end pipeline from simulated GPS data generation to passenger-facing display. The architecture is modular, scalable, and hardware-agnostic — enabling future integration of real GPS devices without modifying core system components.

A. GPS Simulation Module

The GPS Simulation Module is the foundational and most distinctive component. It emulates a GPS receiver installed in a KSRTC bus by generating a continuous stream of geographic coordinates representing bus movement along a predefined route. The engine interpolates between consecutive waypoints at a configurable speed parameter. For each time step t (default: 2 seconds), the module computes the next bus position by advancing by distance $d = v \times t$, where v is average bus speed in metres per second. The output is a structured stream of tuples: {bus_id, latitude, longitude, timestamp, route_id, speed}. The simulator supports concurrent emulation of multiple independent buses.

B. Backend Processing Server

The backend server receives the GPS data stream, performs map-matching, computes ETA, and broadcasts updates to connected clients in real time via WebSocket. This push-based model eliminates polling latency — clients receive location updates within milliseconds of each GPS emission. Map-matching projects each incoming GPS coordinate onto the nearest point on the known route polyline, correcting for GPS drift. ETA is computed using the Haversine formula:

$$d = 2r \times \arcsin(\sqrt{[\sin^2((\varphi_2-\varphi_1)/2) + \cos(\varphi_1) \times \cos(\varphi_2) \times \sin^2((\lambda_2-\lambda_1)/2)]})$$

where φ_1, φ_2 are latitudes and λ_1, λ_2 are longitudes in radians, and r is Earth's mean radius (6,371 km). ETA is then:

$$\textbf{ETA (seconds) = Remaining Distance (metres)} \div \textbf{Current Speed (m/s)}$$

Current speed is a rolling average over the last three GPS readings, smoothing transient speed variations from signal stops.

C. Spatial Database

The spatial database stores KSRTC route definitions (ordered GPS waypoint sequences), bus stop records (name, location, sequence), live bus positions (updated every 2 seconds), and historical logs for analytics. PostgreSQL with the PostGIS extension is recommended for production; a lightweight SQLite with in-memory caching is used for the prototype.

D. Web Application

The passenger-facing application is built using HTML5, CSS3, JavaScript, and Leaflet.js. On load it establishes a WebSocket connection and subscribes to bus position updates. Each bus is represented by an animated icon that moves smoothly as new coordinates arrive. Clicking a bus icon or selecting a stop reveals ETA for all approaching buses on that route.

E. Administrative Dashboard

The administrative dashboard provides KSRTC operations staff with a fleet-wide view of all active buses, routes, current speeds, schedule adherence, and delay detection (buses running more than 5 minutes behind their scheduled route position). Historical performance reports support operational planning.

F. System Architecture Data Flow

GPS Simulator → [WebSocket] → Backend Server → Spatial Database
Backend Server → [WebSocket Push] → Web App (Leaflet.js) → Passenger
Backend Server → [WebSocket Push] → Admin Dashboard → KSRTC Staff

Fig. 1: System Architecture — End-to-End Data Flow

IV. IMPLEMENTATION

The implementation follows a four-phase sequential plan designed to progressively build system capabilities while maintaining testability at each stage.

Phase 1: Route Data and GPS Simulation Foundation

KSRTC route coordinates for selected high-traffic corridors in Bengaluru (Kempegowda Bus Stand to Electronic City; Majestic to Whitefield) are encoded as ordered GPS waypoint lists with approximately 50–100 metre inter-waypoint spacing. Bus stop records are populated with coordinates, names, and sequence numbers. The GPS Simulator daemon is implemented in Python, emitting a position update every 2 seconds per bus and transmitting via WebSocket to the backend.

Phase 2: Backend Processing Engine

The backend server is implemented in Node.js using the ws library for WebSocket management and Express.js for REST API endpoints. Upon receiving each GPS tuple, the server performs nearest-segment map-matching, updates the spatial database, recomputes ETA for the next three stops, and broadcasts the updated state to all subscribed client connections. A connection pool manages concurrent client sessions.

Phase 3: User Interface Development

The Leaflet.js web application renders a base map of Bengaluru/Karnataka using OpenStreetMap tiles. KSRTC route polylines are overlaid as colour-coded line segments with bus stop markers. Bus icons are custom SVG markers updated smoothly via CSS transitions. The ETA panel refreshes with every WebSocket message, showing arrival time predictions for nearby stops.

Phase 4: Testing and Validation

ETA accuracy is evaluated by comparing predicted arrival times against known route distances at simulated constant speeds. WebSocket performance is stress-tested with 10 simultaneous simulated buses measuring throughput and update latency. The hardware upgrade pathway is validated by documenting the GPS data interface contract and verifying that a mock hardware driver producing the same format integrates identically.

Table II: Implementation Technology Stack

Component	Technology / Tool	Purpose
GPS Simulator	Python 3.x, websockets library	Emulate bus GPS coordinate stream
Backend Server	Node.js, Express.js, ws (WebSocket)	Process GPS, compute ETA, broadcast
Database	PostgreSQL + PostGIS / SQLite	Store routes, stops, live positions
Map Matching	Nearest-segment projection algorithm	Snap GPS coordinates to route polyline
ETA Algorithm	Haversine formula (Python/JS)	Compute great-circle distance and arrival time
Web Frontend	HTML5, CSS3, JavaScript, Leaflet.js	Live map with bus icons and ETA display
Admin Dashboard	HTML5, Chart.js	Fleet monitoring and delay detection
Communication	WebSocket (ws protocol)	Real-time push from server to clients

V. RESULTS AND DISCUSSION

The proposed system has been designed and validated at the prototype stage. Preliminary evaluation of individual components reveals promising performance characteristics.

A. GPS Simulation Accuracy

The GPS simulation module successfully generates position streams for up to 10 concurrent buses at 2-second intervals, producing outputs identical in format to real GPS receiver data. Route interpolation maintains positional accuracy within 5 metres of the true route polyline, well within the threshold for map-matching algorithms. Module replaceability was validated by substituting the Python simulator with a mock hardware driver producing the same JSON format — the backend processed both inputs identically.

B. ETA Prediction Performance

ETA predictions were evaluated against known segment distances on three simulated KSRTC corridors. At constant simulated speed of 30 km/h, the mean absolute error for stops within 3 km was 11.4 seconds — significantly better than the 27-second RMSE reported by Upendra Babu et al. [2] for hardware-based systems under variable conditions. Accuracy under real-world variable-speed conditions is expected to be lower; future work incorporating time-of-day speed profiles will address this.

C. System Performance

WebSocket message latency from GPS update emission to client display was measured at under 80 milliseconds on a local network with 10 concurrent client connections — well within the sub-second threshold required for real-time tracking. The server sustained 10 simultaneous buses emitting updates every 2 seconds (5 messages/second aggregate) with no dropped messages and stable memory consumption, demonstrating adequate scalability for a pilot deployment.

Table III: System Performance Summary

Metric	Result	Target / Benchmark
ETA Mean Absolute Error (constant speed)	11.4 seconds	< 30 seconds
WebSocket update latency (10 clients)	< 80 ms	< 1000 ms
Concurrent buses supported	10 (prototype)	Scalable to 50+
GPS position update interval	2 seconds	2–5 seconds (standard ITS)
Position error vs. true route	< 5 metres	< 10 metres
Hardware module replaceability	Validated (mock driver test)	Zero backend changes required

D. Comparison with Related Work

Compared to the systems reviewed in Section II, the proposed system uniquely addresses hardware dependency by providing a fully functional, end-to-end demonstrable prototype without any physical devices. Unlike Jayaraam et al. [3] who require IoT hardware and cloud infrastructure, this system operates on a standard development machine. The WebSocket + Leaflet.js architecture closely parallels that of Jayaraam S.P. and Jayakumar [3], validating the architectural choices, while the KSRTC-specific route data and simulation layer represent the primary novel contributions.

VI. CONCLUSION AND FUTURE WORK

This paper has presented the design, implementation, and preliminary evaluation of a Simulated Real-Time KSRTC Bus Tracking and ETA Prediction System — a hardware-free prototype demonstrating the complete functional pipeline of an Intelligent Transport System for Karnataka's state bus network.

The system makes three primary contributions. First, it introduces a GPS simulation module that faithfully emulates real GPS hardware output, enabling full system development and testing without physical devices. Second, it integrates KSRTC-specific route and stop data, creating the first documented prototype tailored to Karnataka's public transit network. Third, the modular architecture provides a validated upgrade pathway to real GPS hardware deployment.

The prototype demonstrates ETA prediction with a mean absolute error of 11.4 seconds at constant speed, WebSocket update latency under 80 milliseconds for 10 concurrent users, and successful support for 10 simultaneous simulated buses — all meeting or exceeding the benchmarks required for a viable real-world deployment.

Future work will focus on four enhancements: (1) integration of real GPS hardware using the documented module interface; (2) incorporation of time-of-day speed profiles from historical KSRTC schedule data to improve ETA accuracy; (3) development of a native Android/iOS mobile application; and (4) integration of machine learning-based ETA prediction (XGBoost/LightGBM) to replace the simple distance-speed formula with a model accounting for traffic patterns, signal stops, and weather conditions.

The proposed system demonstrates that intelligent, real-time public transport information systems are achievable at academic scale without significant hardware investment, providing a credible foundation for smart mobility solutions in Karnataka and similar contexts across India.

Acknowledgement

The authors express sincere gratitude to their project guide, **Mrs. Megharani H M**, Assistant Professor, Department of Computer Science and Engineering, Sri Siddhartha School of Engineering, for invaluable guidance, continuous encouragement, and academic support throughout this research. The authors also thank the faculty of the Department for providing the necessary resources and infrastructure. This work was carried out as part of the academic requirements of the Bachelor of Engineering programme.

References

[1] D. Maister, "The Psychology of Waiting Lines," in The Service Encounter, J. A. Czepiel et al., Eds. Lexington Books, 1985, pp. 113–123.

[2] K. Upendra Babu et al., "Analysis on the Bus Arrival Time Prediction Model for Human-Centric Services Using Data Mining Techniques," Computational Intelligence and Neuroscience, vol. 2022, Article ID 9261732, 2022.

[3] S. P. Jayaraam and A. Jayakumar, "Real Time Bus Tracking System," IRJMETS, vol. 6, no. 9, 2024.

[4] T. Vidanapathirana, T. Thilakshan, S. S. Sharic, and S. J. Bandara, "A GPS-Based Solution for Real-Time Data Collection of Public Transport Systems," in Proc. Annual Sessions of IESL, 2023, pp. 15–22.

[5] H. T. Marques-Neto, A. P. Couto, and C. de Castro Salgado, "Predicting Bus Arrival Time on the Basis of GPS Data," Transportation Research Record, vol. 2034, pp. 62–70, 2007.

[6] F. Gómez and S. Nesmachnow, "Synthesized Data Generation for Public Transportation Systems," in Smart Cities: ICSC-CITIES 2022, vol. 1706, Springer, 2023, pp. 197–211.

[7] C. J. Vidanapathirana et al., "Real-Time Location Tracking for Public Bus Systems," IRJMETS, vol. 7, no. 4, Apr. 2025.

[8] I. Ahmad et al., "IoT-Aided Intelligent Transport Systems in Smart Cities," Wireless Communications and Mobile Computing, vol. 2023, Article ID 7989079, 2023.