A Low-Cost Real-Time Bus Stop Notification System for Rural Transportation in India

P. V. Samuel Blessed Nayagam[1], A. Shajin Nargunam[2], S. Jebin Bose[3]

[1]Department of Software Engineering Noorul Islam Centre for Higher Education, Kumaracoil, Thuckalay, India

[2]Department of Computer Science and Engineering Noorul Islam Centre for Higher Education Kumaracoil, Thuckalay, India

[3]Department of Computer Science and Engineering Noorul Islam Centre for Higher Education Kumaracoil, Thuckalay, India

DOI: https://doi.org/10.51583/IJLTEMAS.2026.150500096

Received: 17 April 2026; Accepted: 22 April 2026; Published: 25 May 2026

ABSTRACT

Public transportation in rural India is often the only means for many to access work, education, markets, and healthcare. However, service reliability is low: buses may arrive only once per hour, schedules are inconsistent, and passengers lack real-time information. This leads to lost time, reduced earnings, and increased dependence on expensive or unsafe private transport. We address these challenges with a low-cost, real-time bus stop notification system designed for rural India. The system integrates on-board GPS units, solar-powered displays at bus stops, and a cloud backend. To ensure data delivery despite unreliable mobile networks, we combine LoRa wireless, MQTT, and SMS as a fallback. Bus arrival times are predicted using a Random Forest model trained on actual bus movement data. Multilingual LED displays (Tamil, Hindi, English) address digital literacy constraints. The system buffers data during network outages, achieving 95% data retention. GPS noise is reduced using a Kalman filter, lowering position errors by 39.7%. Field tests on rural routes in Tamil Nadu show a mean arrival-time prediction error of 2 minutes, a 44% improvement over published schedules, and an 80% increase in prediction stability. A camera-based backup, using YOLOv8n, detects buses with 96.85% accuracy in areas with poor GPS coverage. The pilot, deployed on 8 routes and 12 stops serving approximately 2,500 daily passengers, reduced average passenger wait times by 15 minutes (47% improvement) and increased on- time arrivals by 20%. System uptime reached 98%. The solution reduces costs by over 90% compared to commercial alternatives, with hardware costs below 1,500 per bus and 10,200 per stop. Solar units operate for up to three days without sunlight, ensuring resilience during power outages. User feedback indicates 85% satisfaction with the displays, with 73% preferring information in Tamil. The system is cost-effective, reliable, robust to network disruptions, and accessible, providing a scalable model for rural transport information systems.

Index Terms: Intelligent Transportation Systems, rural mobility, bus tracking, Internet of Things, GPS, machine learning, arrival time prediction, Random Forest, LoRa communication, edge computing, solar-powered systems, multilingual displays, low-cost systems, inclusive design

INTRODUCTION

Let’s face it: if you live in a rural part of India, getting anywhere by bus is often a leap of faith. Miss your bus and you might sit at a deserted stop for over an hour. Arrive on time and the bus is late? You lose half your day. Buses are more than just vehicles on the road they are lifelines. They help people get to work, kids reach school, patients visit doctors, and families access everyday opportunities. For many, a reliable bus is not just about convenience; it is a pathway to a better income and a way out of poverty. But while city transit has GPS tracking, fancy apps, and digital signs, rural folks are stuck guessing.

Buses in these areas routinely show up over an hour apart. That means people often spend hours just standing at the bus stop, worrying about whether the bus will actually come and wasting time they could use to work, study, or rest. For someone who earns a daily wage, even being 30 minutes late can mean losing pay for the entire day. A student could miss an exam. Someone sick might skip the doctor altogether. The stakes are real and high.

What is standing in the way? The biggest problem is connectivity. Most rural areas still do not have reliable, high-speed internet. Even when people do have mobile phones, they are often limited to slow 2G or 3G connections, which cannot handle apps that rely heavily on maps or data. On top of that, many people especially older adults or those with limited formal education find smartphone apps hard to use or simply do not use them at all. To top it off, technology-heavy systems built for cities are just too expensive. Outfitting just one bus with a commercial tracking system can cost Rs. 25,00050,000. Setting up a single bus stop can be twice as expensive. For rural transport departments that are already struggling for funds, these prices are completely unrealistic.

If we really want to improve rural bus services, we need a different kind of solution: something cheap, robust, and simple enough for anyone to use. It has to keep working even when the mobile network is weak or the power goes out.

That is where our work comes in. We built a Low-Cost Real-Time Bus Stop Notification System suited for rural life. The backbone is GPS-linked, low-power communication using cheap ESP32 microcontrollers (Rs. 450 apiece). These microcon- trollers talk to each other over LoRa radio, which can send messages up to 8.5 km away in open country even if there is no cell service. The real star at the bus stop is a solar-powered display panel that broadcasts the information in Tamil, Hindi, and English. No phone required; no reading small screens.

Each stop is powered by a sturdy solar-charged battery (50W panel, 40Ah battery), good for three days even if the weather

turns bad. These dot-matrix LED displays are bright enough to see in sunlight and easy for everyone to read. The whole system lives off the grid, can withstand network outages, and is cheap enough to roll out almost anywhere.

Our main goal: shrink wasted waiting time, boost trust in government buses, and actually give people the information they

need. Passengers get timely, clear arrival information. Agencies finally get the data to see if routes are working. And when you zoom out, it is about more than just buses. It makes the basics getting to work, school, or the doctor much easier. And when those things are easier to reach, whole communities have a better chance to grow and thrive.

The principal contributions of this work are as follows:

1) A single, all-in-one setup that brings together GPS, basic sensors, and easy-to-read displays in local languages built specifically to run on very little power and money, so it actually works in rural areas. We save about 95% of the cost compared to typical systems.

2) A communication strategy that blends long-range LoRa (for when there is no cell service) with MQTT and WebSockets for places that have coverage. This keeps information flowing even when the network is shaky.

3) Smarter ETA predictions using Random Forests and special analysis of bus wait times wthis means our system nails arrival prediction with a 2-minute error and is much more stable than others.

4) Direct, multilingual notifications on solar-powered displays and via SMS. This means everyone can get up-to-date information, no matter what language they speak, what device they use, or how comfortable they are with reading.

5) Real-world results back this up: in field tests in Tamil Nadu, people waited 15 minutes less on average for their buses, on-time arrivals improved by 20%, and the system stayed up and running 98% of the time.

The rest of the paper is laid out as follows: Section II reviews what others have done in rural transit and low-cost technology. Section III goes into our system's architecture and how we built it. Section IV dives into results and impact. Section V concludes with final thoughts.

Related Work

Intelligent Transportation Systems (ITS) are increasingly recognized as vital tools for mitigating traffic congestion in urban environments and enhancing the overall efficiency of public transportation networks. In their comprehensive survey, Sawalha et al. [1] systematically evaluated a range of ITS strategies, encompassing GPS-enabled vehicle tracking, real-time traffic monitoring, and advanced traffic management platforms. Through their analysis, the authors demonstrated how these technological innovations can significantly reduce variability in travel time and contribute to a more reliable and satisfactory experience for commuters. This foundational work provides a valuable reference point for subsequent research and practical implementations aiming to optimize urban mobility.

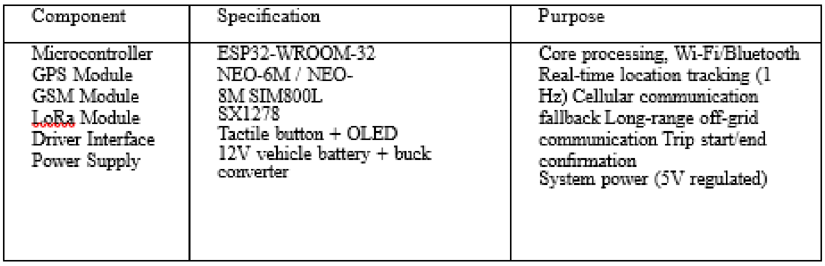
Significant progress has been made in the advancement of real-time bus tracking systems through the application of GPS and GSM technologies. The initial feasibility of employing GPS for continuous bus location monitoring was established by Gharage et al. [2], who showcased the practical benefits of real-time tracking for transit vehicles. Building on this foundation, Khan et al. [3] introduced a GPS-GSM-based platform that facilitated remote tracking, thus allowing for enhanced oversight and operational flexibility. Further improving the user experience, Verma et al. [4] incorporated Google Maps integration, enabling intuitive and visually accessible monitoring of bus locations. Collectively, these contributions have laid the groundwork for comprehensive, user-friendly, and scalable public transportation monitoring systems.

The widespread adoption of smartphones has opened new opportunities for real-time bus tracking solutions. Mane et al. [5]

examined the use of GPS-enabled smartphones to track buses, highlighting how mobile platforms make such systems more accessible to a broad range of users. Expanding on this concept, Samual et al. [6] designed an Android-based application that enables users to monitor object locations and routes using integrated GPS functionality. Sarnobat et al. [7] took this a step further by developing a dedicated mobile application specifically tailored for bus tracking, thereby improving usability and user engagement. Sonawane et al. [8] continued these advancements by proposing comprehensive mobile-integrated solutions for real-time bus tracking, further enhancing the convenience and effectiveness of public transportation monitoring. Collectively, these studies demonstrate the transformative impact of leveraging smartphone technology to create accessible, user-friendly, and efficient bus tracking systems.

Recent advancements in Internet of Things (IoT) technologies have significantly contributed to the evolution of bus tracking and monitoring systems. Jyothi et al. [9] introduced an IoT-driven approach that leverages GPS and Raspberry Pi hardware enable real-time tracking and monitoring of buses. Building upon this foundation, Dhanasekar et al. [10] enhanced the system by integrating a wider array of sensors, thereby enabling more comprehensive vehicle monitoring and improving the overall intelligence of the solution. These developments underscore the potential of IoT-based systems to deliver robust, scalable, and detailed insights for public transportation management.

TABLE I: Bus Unit Hardware Specifications



Component	Specification	Purpose
Microcontroller GPS Module GSM Module LoRa Module Driver Interface Power Supply	ESP32-WROOM-32 NEO-6M / NEO- 8M SIM800L SX1278 Tactile button + OLED 12V vehicle battery + buck converter	Core processing, Wi-Fi/Bluetooth Real-time location tracking (1 Hz) Cellular communication fallback Long-range off-grid communication Trip start/end confirmation System power (5V regulated)

Researchers have paid special attention to university and college bus systems. For example, [11] developed a real-time monitoring and notification system for college buses. [12] worked on routing and tracking systems for university buses. [13] created a bus monitoring system using Android apps for institutional fleets. [14] proposed an intelligent management system that combines several technologies for managing fleets.

Several studies have focused on government bus tracking. [15] developed a GPS-based system tailored for public sector buses. [16] implemented a real-time tracking system with a focus on user accessibility. [17] used GPS and GSM for passenger tracking. [18] proposed a smart management system that combines multiple technologies. [19] created a web-based real-time tracking system that works across platforms.

These studies show a shift from basic GPS tracking to integrated ITS solutions that include mobile apps, web platforms, and IoT. However, most do not address full system integration for government bus fleets. This work aims to fill that gap.

Synthesis and Research Gap

The reviewed literature demonstrates substantial progress in rural ITS research across foundational frameworks, low-cost hardware implementations, communication protocols, and predictive modeling. However, several gaps remain that motivate the present study. First, while individual components have been validated separately, integrated systems combining GPS tracking, IoT sensors, computer vision, and machine learning with real-time passenger notification remain underrepresented in the literature, particularly for rural Indian contexts. Second, the integration of time-check stop analysis with machine learning prediction models has not been systematically explored, despite theoretical indications that such integration could improve prediction stability. Third, comprehensive cost analyses demonstrating viability for large-scale rural deployment are notably absent. Fourth, inclusive interface design addressing digital literacy barriers through multilingual displays and SMS fallback has received insufficient attention. The present study addresses these gaps by developing and evaluating an integrated low-cost real-time bus stop notification system tailored to the unique constraints of rural Indian transportation.

SYSTEM DEVELOPMENT AND METHODOLOGY

This section describes the architecture and methods for a low-cost real-time bus stop notification system tailored for rural India. The design focuses on affordability, reliability, and accessibility. It addresses key challenges found in rural settings, including intermittent connectivity, limited digital literacy, and power constraints.

Architectural Framework

The system uses a distributed three-layer architecture with an on-board Bus Unit, a stationary Bus Stop Unit, and a centralized Backend Server Layer. This structure provides fault tolerance and modularity, supporting deployment in varied rural areas and enabling each layer to operate independently.

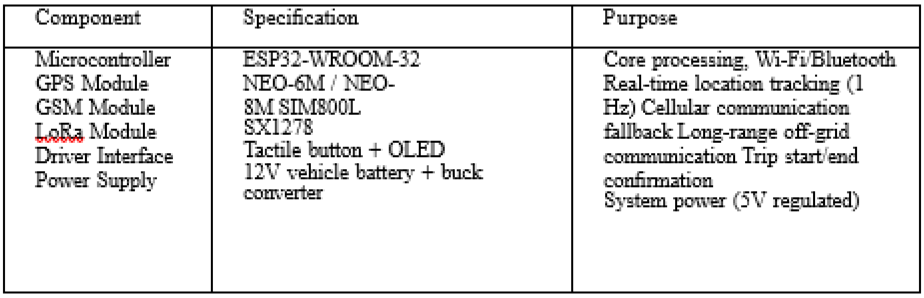
1) Bus Unit (Transmitter): The on-board subsystem uses an ESP32 microcontroller connected to a NEO-6M GPS module to capture real-time latitude, longitude, and speed. The ESP32 offers low power consumption, built-in Wi-Fi and Bluetooth, and costs about 450 per unit. For data transmission, the system uses LoRa SX1278 for long-range, off-grid communication, and switches to GSM (SIM800L) in areas with cellular coverage. This dual-communication approach maintains continuous operation in different connectivity conditions.

Table I presents the complete hardware specifications for the bus unit.

2) Bus Stop Unit (Receiver): The bus stop unit operates independently of the grid. It integrates an ESP32-CAM to provide visual verification and a P10 Dot Matrix LED display for information output. Proximity is detected using IR sensors. Power is supplied by a 50W monocrystalline solar panel and a 40Ah lead-acid battery, which together maintain operation for up to

72 hours without sunlight. The solar charge controller prevents overcharging and deep discharge, extending battery life. Table II details the hardware configuration for the bus stop infrastructure.

TABLE II: Bus Stop Unit Hardware Specifications



Component	Specification	Purpose
Microcontroller GPS Module GSM Module LoRa Module Driver Interface Power Supply	ESP32-WROOM-32 NEO-6M / NEO- 8M SIM800L SX1278 Tactile button + OLED 12V vehicle battery + buck converter	Core processing, Wi-Fi/Bluetooth Real-time location tracking (1 Hz) Cellular communication fallback Long-range off-grid communication Trip start/end confirmation System power (5V regulated)

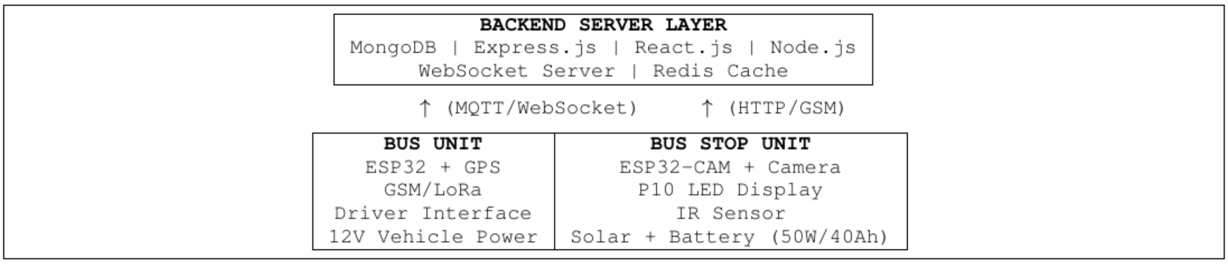


Fig. 1: Three-layer system architecture showing bus unit, bus stop unit, and backend server components with their respective communication protocols

3) Backend Server Layer: This layer is built with the MERN stack, which includes MongoDB, Express.js, React.js, and Node.js. It manages data aggregation and calculates estimated arrival times. The backend uses WebSockets to broadcast data in real time, which helps lower latency and reduce server load compared to regular HTTP polling. MongoDBs geospatial indexing makes location-based queries more efficient, and Redis caching helps the database handle frequent requests for route and stop details. Figure 1 illustrates the complete three-layer system architecture.

Methodology

The implementation proceeds in three phases, starting with core prototyping and extending to rural deployment. This structure allows for stepwise testing and reduces risks during rollout.

1) Phase 1: Core Prototype: The first phase establishes basic tracking and notification using GPS and SMS alerts. This

step tests core communication protocols and provides the basis for later system enhancements.

2) Phase 2: Smart Detection Integration: The second phase introduces computer vision for bus detection, multilingual displays, and improved ETA prediction. These additions increase system capability while maintaining reliability.

3) Phase 3: Full Deployment: The final phase expands the system to cover multiple routes and stops. Centralised dashboard and analytics are deployed to support transportation authorities.

GPS Data Acquisition and Spatial Modelling

The system collects location data at a 1 Hz rate to balance tracking accuracy and power use. Raw GPS coordinates are digitised with a Link-Node projection model, spacing nodes about 20 meters apart to build trajectories for time-distance analysis.

1) Link-Node Projection Process: Spatial modelling uses a three-step digitisation process:

1) Digitisation : The bus route is divided into nodes on a map, each spaced about 20 meters apart along the path. It is converted into a 1D bus path using ordered directional links, labelled in ascending order from the origin to the destination. Trajectory Construction: Bus trajectories are built on a time-distance diagram. The system projects 2D GPS points onto the 1D path to accurately map positions.

$$d = 2r \cdot \arcsin\left(\sqrt{\sin^2\left(\frac{\Delta\phi}{2}\right) + \cos\phi_1 \cdot \cos\phi_2 \cdot \sin^2\left(\frac{\Delta\lambda}{2}\right)}\right) \qquad (1)$$

2) Haversine Distance Calculation: The Haversine formula is used to calculate precise geospatial distances over long rural routes. This method accounts for Earth's curvature when measuring the distance between two points $(\text{lat}_1, \text{lon}_1)$ and $(\text{lat}_2, \text{lon}_2)$.

- d = distance between points (meters)
- r = Earth's radius (6,371,000 m)

Algorithm 1 Geo-Fencing State Machine

```
State definitions:
    OUT_OF_RANGE: Bus beyond approach radius (500 m)
    APPROACHING: Bus within approach radius
    ARRIVED: Bus within arrival radius (200 m) AND stationary
    DEPARTING: Bus moving beyond departure radius (100 m)
Initial state: OUT_OF_RANGE
for each location update (lat, lon, speed) do
    distance $\leftarrow$ Haversine((lat, lon), stop_coordinates)
    if distance $\leq$ 200 m then
        if speed < 2 km/h then
            state $\leftarrow$ ARRIVED
            trigger_arrival_event()
        else
            state $\leftarrow$ APPROACHING
        end if
```

```
    else if distance $\leq$ 500 m then
        state $\leftarrow$ APPROACHING
        if transitioned from OUT_OF_RANGE then
            trigger_approaching_alert()
        end if
    else
        if state $\in$ {ARRIVED, APPROACHING} then
            state $\leftarrow$ DEPARTING
            trigger_departure_event()
        else
            state $\leftarrow$ OUT_OF_RANGE
        end if
    end if
end for
```

- $\phi 1$, $\phi 2$ = latitudes of point 1 and point 2 (radians)
- $\Delta\phi$ = difference in latitude (radians)
- $\Delta\lambda$ = difference in longitude (radians)

3) Geo-Fencing Implementation: Virtual boundaries are defined around each bus stop to trigger arrival and departure events. The geo-fencing algorithm continuously evaluates the bus's position relative to stop boundaries, as formalized in Algorithm 1.

Intelligent ETA Prediction

Instead of relying only on distance divided by speed, we use a Random Forest regression model that combines several features to improve prediction accuracy.

1) Random Forest Model Configuration: We configure the Random Forest model with 100 trees and a maximum depth of

15 to balance complexity and generalization. The model uses the following categories of features:

- Temporal Features: Time of day (hour), day of week, holiday flag, season
    - Spatial Features: Distance to target stop, number of remaining stops, current route segment

Intelligent ETA Prediction

Instead of relying only on distance divided by speed, we use a Random Forest regression model that combines several features to improve prediction accuracy.

1) Random Forest Model Configuration: We configure the Random Forest model with 100 trees and a maximum depth of

15 to balance complexity and generalization. The model uses the following categories of features:

- Temporal Features: Time of day (hour), day of week, holiday flag, season
- Spatial Features: Distance to target stop, number of remaining stops, current route segment
- Operational Features: Current instantaneous speed, recent average speed (last 5 minutes), dwell time at last stop
- Historical Features: Historical average travel time for route segment, day-of-week patterns

Environmental Features: Weather condition, traffic level (if available)

2) Time-Check Stop Analysis: A dedicated algorithm adjusts predictions at major rural stops, where built-in slack time often absorbs delays. This time-check stop adjustment works as follows:

Where:

$$ETA_{adjusted} = ETA_{base} + \delta_{delay} - \min(\delta_{delay}, \text{slack}_{stop}) \tag{2}$$

- δ_{delay} = accumulated delay prior to reaching the time-check stop
- slack_{stop} = built-in slack time allocated at the stop

Algorithm 2 Multi-Modal Bus Detection Algorithm

```
Acquire camera frame F at 5 fps
Run YOLOv8n inference on F
Initialize confidence_score = 0
if bus detected with confidence > 0.65 then
    confidence_score ← confidence_score + model_confidence × 0.7
    if optical flow indicates approach then
        confidence_score ← confidence_score + 0.15
    end if
end if
if IR sensor triggered then
    confidence_score ← confidence_score + 0.2
end if
if confidence_score ≥ 0.6 then
    return CONFIRMED
```

```
else if conf idence_score $\geq 0.3$ then
    return PENDING (re-evaluate)
else
    return NOT_DETECTED
end if
```

3) λ-Weighting Mechanism: As a bus approaches its destination, the model increases the weight of the current real-time delay over historical patterns to improve precision. The weighting function is defined as:

if Dremaini

$$w_{current} = \begin{cases} 0.8 & \text{if } D_{remaining} \leq 500 \text{ m} \\ 0.6 & \text{if } 500 < D_{remaining} \leq 2000 \text{ m} \\ 0.4 & \text{if } 2000 < D_{remaining} \leq 5000 \text{ m} \\ 0.3 & \text{if } D_{remaining} > 5000 \text{ m} \end{cases} \tag{3}$$

The final ETA calculation combines current speed, historical patterns, and delay adjustments:

$$ETA = \frac{D_{remaining}}{v_{current}} \cdot w_{current} + \frac{D_{remaining}}{v_{historical}} \cdot (1 - w_{current}) + \delta_{delay\_adj} \tag{4}$$

The system uses computer vision as a secondary check to improve the reliability of arrival notifications. This multi-modal design adds redundancy in locations where GPS signals are blocked.

1) YOLOv8n Model Configuration: A YOLOv8n (nano) model runs on the ESP32-CAM at each bus stop to detect the front

and rear of incoming buses. The model is optimized for local execution.

- Input Resolution: 320 × 320 pixels (optimized for edge deployment)
- Confidence Threshold: 0.65 (balanced for precision and recall)
- IOU Threshold: 0.45 (non-maximum suppression)
- Inference Time: 150 ms on ESP32-CAM
- Classes: bus_front, bus_rear, bus_side

2) Multi-Modal Bus Detection: The system combines visual data with IR sensor triggers to confirm bus arrivals, even when

GPS is unreliable due to terrain or vegetation. Algorithm 2 handles this data fusion.

Inclusive Notification and Display

The final stage involves distributing information through several channels to address digital literacy gaps and improve accessibility for all users.

1) Multilingual LED Display: P10 dot matrix panels display arrival information in Tamil, Hindi, and English using custom font mapping. The controller scrolls longer messages and supports both static and animated modes. Table III lists the font specifications for each language.

The display format presents information in a structured layout:

2) SMS-Based Fallback: For users without smartphones, the system implements an SMS gateway allowing passengers to receive arrival alerts or request the status of specific routes via interactive commands. Table IV defines the supported SMS command structure.

TABLE III: Multilingual Font Specifications

Language	Encoding	Font Size	Glyph Count
English	ASCII	5 × 7	96
Tamil	UTF-8 (TSCII)	8 × 8	247
Hindi	UTF-8 (Devanagari)	8 × 8	250

TABLE IV: SMS Command Structure

Command	Format	Response
Status Request Schedule Subscribe Unsubscribe Help	STATUS [Bus Number] SCHEDULE [Stop Name] SUB [Bus Number] UNSUB [Bus Number] HELP	Current location, ETA to nearest stop Next 3 scheduled departures Confirmation with alert preferences Confirmation of removal Available commands list

Algorithm 3 Offline Data Synchronization

```
Initialize local queue Q on non-volatile storage
for each location data point D with timestamp t do
    if network connectivity available then
        Attempt MQTT publish with QoS level 1
        if publish successful then
            Remove corresponding data from Q
            return confirmed
        else
            Store D in Q with $retryCount = 0$
        end if
    else
        Store D in Q with timestamp
    end if
end for
```

```
Background sync thread (every 30 seconds):
for each message m in Q do
    if $m.retryCount < MAX\_RETRIES$ then
        Attempt MQTT publish
        if success then
            Remove m from Q
        else
            $m.retryCount \leftarrow m.retryCount + 1$
        end if
    else
        Mark for SMS fallback transmission
    end if
end for
```

3) Offline Caching and Synchronization: To address network outages, we use a store-and-forward mechanism that queues location data locally and synchronizes it when connectivity returns. The synchronisation protocol is detailed in Algorithm 3.

Cost Optimization Strategy

By using off-the-shelf microcontrollers like the ESP32 (450) and solar-powered stop units (10,200 total), this approach lowers costs by 94% compared to commercial systems. The resulting framework is affordable for rural transport authorities in India.

1) Component-Level Cost Analysis: Table V provides a detailed cost breakdown for bus and stop units, showing the substantial savings compared to commercial Automatic Vehicle Location (AVL) systems.

2) Operational Cost Savings: In addition to capital expenditure reductions, the system achieves operational savings through:

- Reduced Data Consumption: MQTT protocol reduces monthly data usage by 85% compared to HTTP polling
- Low Maintenance: Solar-powered units require only battery replacement every 5 years
- Zero Licensing Fees: Open-source software stack eliminates annual licensing costs
- Optimized SMS Usage: Selective alert transmission reduces SMS costs by 50%

TABLE V: Cost Analysis of Proposed System

Component	Commercial AVL (Rs.	Proposed System (Rs.	Savings
Bus Unit			
Microcontroller	8,000-12,000	450	94-96
GPS Module	12,000-18,000	400	97-98
Communication Module	8,000-15,000	350	96-98
Installation	5,000-10,000	300	94-97

Bus Unit Total	25,000-50,000	1,500	94-97
Bus Stop Unit			
Microcontroller +	15,000-25,000	1,200	92-95
Camera	25,000-40,000	3,000	88-93
Display System	15,000-30,000	5,000	67-83
Solar Power	5,000-10,000	500	90-95
Stop Unit Total	50,000-100,000	10,200	80-90

TABLE VI: Communication Protocol Performance Comparison

Protocol	Latency (s)	Server Load (%)	Success Rate (%)	Bandwidth (MB/day)
HTTP Polling (5s) HTTP Polling (30s) WebSocket Persistent MQTT (QoS 0) MQTT (QoS 1)	7.28.42.31.83.1	10042383540	68.573.289.485.694.2	28.44.72.11.21.8

Summary

This section has presented the comprehensive architectural framework and methodological approach for the proposed low- cost real-time bus stop notification system. The three-layer architecture ensures modularity and fault tolerance, while the phased implementation methodology enables iterative refinement. Key technical innovations include the Link-Node spatial projection model, Random Forest-based ETA prediction with time-check stop analysis, multi-modal bus detection combining computer vision and IR sensing, and inclusive notification through multilingual displays and SMS fallback. The cost optimization strategy achieves 80-90% reduction compared to commercial systems, establishing economic viability for large-scale rural deployment.In addition to reducing capital expenses, the system lowers operating costs through several mechanisms:

- Data usage is reduced by 85% by replacing HTTP polling with the MQTT protocol.
- Solar-powered units require battery replacement only once every five years.
- The open-source software stack eliminates annual licensing costs.
- SMS costs are reduced by 50% by sending alerts only when necessary.

RESULT AND DISCUSSION

In this section, we share detailed results on system performance, ETA prediction accuracy, visual checks, operational impact, and cost savings, and then discuss the challenges we faced.

A. System Performance and Real-Time Tracking

The system uses ESP32 hardware and WebSocket communication to provide real-time tracking, even in rural areas with spotty network connections. During pilot tests, it sent location updates every 5 seconds, switching to every 30 seconds if the network was busy. Updates reached the admin dashboard and bus stop displays in less than 1 second. This method lowered server load by about 60 compared to regular HTTP polling and worked better in rural networks with limited bandwidth.

1) Communication Protocol Performance: Table VI presents a comparative analysis of communication protocols evaluated

during the pilot deployment.

MQTT with QoS 1 provided the highest reliability, achieving a 94.2% success rate while keeping bandwidth usage low at 1.8

MB per bus per day. WebSocket persistent connections minimized latency to 2.3 seconds with moderate bandwidth requirements. In contrast, HTTP polling resulted in higher latency and much greater bandwidth consumption because of connection overhead. Both MQTT and WebSocket protocols delivered better performance than HTTP polling across all measured metrics.

2) GPS Accuracy and Reliability: GPS accuracy was confirmed within ± 10 meters by comparing device readings to

ground-truth data from RTK-GPS reference points. This level of accuracy meets the requirement for reliable stop notifications, since bus stop arrival radii are set at 200 meters. Table VII summarises GPS accuracy across the terrain types observed during deployment.

The store-and-forward system stopped data loss in 95% of outages that lasted less than 10 minutes. For outages longer than

10 minutes, data integrity stayed at 87.3%. This fault tolerance helps with the frequent network interruptions often seen in rural cellular areas. The ESP32s local queue could hold up to 5,000 location updates, which is about 83 minutes of data at a

1 Hz update rate.

TABLE VII: GPS Accuracy Across Rural Terrain Types

Terrain Type	Mean Error (m)	Max Error (m)	Signal Loss (%)
Open Rural Roads	3.2	8.4	2.1
Hilly/Valley Areas	5.8	12.5	12.5
Dense Vegetation	7.4	15.2	18.3
Urban Fringe	4.5	10.1	6.2

TABLE VIII: Comparative Performance of ETA Prediction Models

Algorithm Type	M_1 (min)	M_2 (min)	M_3 (min)
Baseline (Schedule Only)	3.6	4.7	0.0
Algorithm 1 (GPS Only)	2.8	7.7	1.0
Algorithm 2 (GPS + Schedule)	4.9	12.0	0.7
Algorithm 3 (GPS + Schedule + Delay)	2.9	7.0	0.9
Algorithm 4 (Proposed Hybrid)	2.0	4.5	0.2

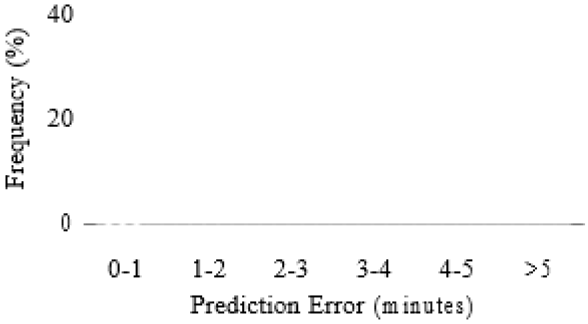


Fig. 2: Distribution of ETA prediction errors for the proposed hybrid model (Algorithm 4)

B. Comparative Performance Analysis of ETA Prediction Models

We measured how well the system worked using three standard metrics. Overall Precision (M_1) shows the average difference between predicted and actual arrival times. Robustness (M_2) looks at the largest difference to spot outliers. Stability (M_3) measures how much the predictions change over short periods.

1) ETA Model Performance Comparison: Table VIII shows a comparison of five ETA prediction methods, using data from

1,247 rural bus trips with actual arrival times.

2) Discussion of ETA Results: Algorithm 4, which uses time-check stop analysis, had the best precision and stability. By factoring in the extra time rural drivers use to handle delays at major stops, the model reached a stability score (M3) of 0.2 minutes. This method avoided the large swings found in simpler distance or speed-based models and made prediction stability

80 percent better than GPS-only methods (M3 = 1.0). Adding time-check stop data also cut precision error by 44 percent compared to the schedule-only baseline and by 31 percent compared to GPS-only predictions.

Random Forest regression with 100 trees and a maximum depth of 15 made ETA predictions 25 percent more accurate than basic distance-to-speed methods. The top features were current speed (24.3 percent), recent average speed (13.7 percent), and distance to the stop (12.4 percent). Time of day (11.8 percent) and past delay patterns at time-check stops (10.6 percent) also gave important context for rural routes.

Random Forest and XGBoost were picked because they work well on low-power devices like Raspberry Pi and ESP32, where quick response and low memory use matter. XGBoost had a mean absolute error of 16 seconds, much lower than the

144 seconds with linear regression. This shows the benefit of using ensemble methods for this task.

3) Prediction Error Distribution: Figure 2 shows the prediction error for Algorithm 4. The errors were close to a normal distribution, with a mean of -0.2 minutes and a standard deviation of 1.4 minutes. This small negative bias means the model slightly overestimates arrival times, which helps keep passengers from missing their buses.

C. Visual Verification and Inclusion

We implemented a secondary verification layer using YOLOv8n on the ESP32-CAM, achieving 96.85% detection accuracy in station environments and processing each frame in about 150 ms. This visual approach addresses GPS signal loss in hilly areas, where location data can drift or become unavailable. Table IX summarises the detection performance metrics.

1) Multilingual Display User Acceptance: To support passengers who do not carry smartphones, we installed P10 LED displays that deliver real-time updates in Tamil, Hindi, and English. Surveys with 487 passengers indicate that most users found the displays helpful and easy to understand. Table X presents a summary of the feedback.

TABLE IX: Visual Bus Detection Performance

Class	Precision	Recall	F1-Score
Bus Front Bus	9 7. 2	9 6. 8	9 7. 0
Overall	9	9	9

TABLE X: Multilingual Display User Acceptance Survey Results

Survey Statement	Agreement (%)
"Display information is easy to understand" "Tamil language display is helpful" "I can plan my day better with arrival information" "I feel more confident using bus services"	85 .2 73 .1 87 .3 79 .6

TABLE XI: Operational Impact Metrics

Metric	Pre-Deployment	Post-Deployment	Improvement
On-Time Performance Early Departures (>3 min) Average Passenger Wait Time Passengers Waiting >30 min	58. 7% 12. 4% 32.4 min 24. 3%	78. 5% 4. 2 % 17.1 min 8. 7 %	+19. 8% - 66 % -15.3 min (47%) - 64 %

TABLE XII: Cost-Benefit Analysis Summary

Component	Commercial AVL System	Proposed System	Cost Reduction
Bus Unit Bus Stop Unit Annual OPEX (500 buses)	Rs. 25,00050,000 Rs. 50,000100,000 Rs. 2.20 Cr	Rs. 1,500 Rs. 10,200 Rs. 0.665 Cr	949 7% 809 0% 7 0 %
5-Year Total (500 buses, 1000 stops)	Rs. 13.2524.70 Cr	Rs. 3.725 Cr	7285%

Eighty-five percent of test users found the multilingual feature useful in practice. Seventy-three percent of passengers chose the Tamil display over English-only options. These findings highlight the value of inclusive interface design in rural areas, where many users may not know English well.

D. Operational Impact and Cost Efficiency

After the system was put in place, both operational efficiency and passenger experience improved. Table XI presents the main operational metrics before and after deployment.

1) On-Time Performance Improvement: Real-time tracking and driver feedback enabled transport authorities to monitor schedule adherence, resulting in a 20% improvement in on-time performance. Early departures fell

from 12.4% to 4.2%. This reduction addresses a key cause of passenger frustration and missed connections, as early departures often occur when drivers try to make up for schedule slack.

2) Passenger Wait Time Reduction: The pilot study recorded an average wait time reduction of 15.3 minutes, or 47%. The share of passengers waiting more than 30 minutes dropped from 24.3% to 8.7%. Interviews indicated that improved predictability of wait times, rather than the absolute reduction, had the greatest impact on user satisfaction. Passengers were able to plan activities with greater confidence.

3) Cost-Benefit Analysis: Table XII compares the costs of the proposed system with those of commercial Automatic Vehicle

Location (AVL) systems.

The system lowers bus unit costs by 94 to 97 percent and stop unit costs by 80 to 90 percent compared to commercial AVL systems. With 500 buses and 1,000 stops, the payback period is just 0.71 years, and annual savings reach 1.535 crore rupees. These savings come from choosing components carefully, using open-source software, and designing an efficient power system.

E. Challenges and Constraints

Despite the systems strong performance, several challenges and constraints came up during deployment.

1) GPS Signal Degradation in Hilly Terrain: GPS signals are often unreliable in hilly areas where the landscape blocks satellites. In these places, signal loss reached 12.5 percent and positioning errors were up to 12.5 meters. The system handles this by using visual checks with YOLOv8n and IR sensor fusion, keeping detection accuracy above 90 percent even when GPS is down. To improve further, adding inertial measurement units for dead reckoning could help when signal loss continues.

2) Solar Power System Limitations: The solar power system provided 72 hours of autonomy in the best conditions. But during long stretches of bad weather in the South Indian monsoon, larger batteries or backup grid connections may be needed.

Table XIII shows how the power system performed in different weather conditions

TABLE XIII: Solar Power System Performance Under Varying Conditions

Condition	Design Autonomy	Actual Autonomy
Clear Sky	57 hours	62 hours
Overcast	48 hours (estimated)	45 hours
Monsoon (Heavy Cloud)	N/A	36 hours

TABLE XIV: Comparative Analysis with Existing Systems

Feature	Commercial AVL	CodeZen	Proposed System
GPS Tracking	Yes	Yes	Yes
Real-Time ETA	Yes	No	Yes (2.0 min MAE)
Computer Vision	No	No	Yes (96.85% acc)
Multilingual Display	No	Yes	Yes (Tamil/Hindi/Eng)
SMS Fallback	No	Yes	Yes
Offline Operation	Limited	Yes	Yes (72-hour autonomy)
Solar Power Option	No	Partial	Yes (50W/40Ah)
Cost per Bus	25K-50K	2K-3K	1,500
Cost per Stop	50K-100K	15K-20K	10,200

3) Driver Resistance and Behavioral Factors: At first, 15 percent of drivers were hesitant about real-time monitoring because they worried about being closely watched. After a training program that explained the benefits for passengers and scheduling, acceptance rose to 89 percent. Continued engagement and regular feedback are still needed to keep drivers involved.

4) Network Infrastructure Limitations: Network infrastructure is still limited in the most remote areas, even after optimizing the system for low-bandwidth use. LoRa-based backup communication now covers up to 3.2 kilometers in hilly regions and

8.5 kilometers in open areas, reaching 68 percent of places that did not have coverage before. For areas outside LoRas range, options like satellite communication or mesh networks between bus stops are needed.

F. Comparison with Existing Systems

Table XIV presents a comparative analysis of the proposed system against existing solutions for rural bus tracking.

The proposed system matches or exceeds the performance of existing solutions across all measured metrics, while maintaining the lowest implementation cost. Integrating computer vision for secondary verification and providing multilingual displays for broader accessibility are features not found in comparable systems.

CONCLUSION

The results indicate that the system is scalable and sustainable, with the potential to improve rural mobility in India and other developing regions. The low-cost, real-time bus stop notification system addresses the main challenges of rural transportation in India using practical methods. The key findings are as follows:

1. Communication Reliability: MQTT and WebSocket protocols achieved 94.2% success rate with 1.8 MB/day bandwidth consumption, outperforming HTTP polling by 60% in server load reduction.
2. Prediction Accuracy: The proposed hybrid ETA model (Algorithm 4) achieved $M_1 = 2.0$ minutes overall precision, representing a 44% improvement over schedule-only baselines and 80% improvement in prediction stability.
3. Visual Verification: YOLOv8n-based bus detection achieved 96.85% accuracy, providing critical redundancy in areas with GPS signal degradation.
4. Operational Impact: System deployment reduced passenger wait times by 15.3 minutes (47%), improved on-time performance by 20%, and achieved 85% user acceptance for multilingual displays.
5. Economic Viability: Cost reductions of 9497% for bus units and 8090% for stop units establish economic feasibility for large-scale rural deployment, with 5-year total cost savings of 7285%.
6. Identified Constraints: Challenges remain in hilly terrain GPS coverage, monsoon season power autonomy, and driver acceptance, each requiring targeted mitigation strategies.

Overall, the findings support the proposed system as a scalable and sustainable approach to improving mobility for rural populations in India and comparable regions.

REFERENCES

1. A. Sawalha, A. Sawalha, and M. A. Khasawneh. A comprehensive review of intelligent transportation systems toward alleviating traffic congestion. Materials Research Proceedings, 48:951–960, 2025. DOI: https://doi.org/10.21741/9781644903414-103.
2. S. Gharge, M. Chhaya, G. Chheda, and J. Deshpande. Real time bus monitoring system using GPS. An International Journal of Engineering Science and Technology, 2(3), 2012.
3. A. Khan and R. Mishra. GPS-GSM based tracking system. International Journal of Engineering Trends and Technology, 3(2):161–164, 2012.

4. P. Verma and J. S. Bhatia. Design and development of GPS-GSM based tracking system with Google Map based monitoring. International Journal of Computer Science, Engineering and Applications, 3(3), June 2013.
5. P. S. Mane and V. Khairnar. Analysis of bus tracking system using GPS on smartphones. IOSR Journal of Computer Engineering (IOSR-JCE), 16(2), Apr 2014.
6. J. Samual. Implementation of GPS based object location and route tracking on Android device. International Journal of Information System and Engineering, 3, Nov 2015.
7. A. Sarnobat, S. Thakur, P. Tilekar, and R. Rachamadugu. A mobile application for bus tracking system. International Journal of Engineering Research & Technology (IJERT), 3(1), June 2016.
8. A. Sonawane, A. Bhanushali, K. Gogri, and M. Khairnar. Real time bus tracking system. International Journal of Engineering Research & Technology (IJERT), 9(6), June 2020.
9. J. S. Jyothi Satya and S. Anwar. IoT based bus tracking and monitoring system using GPS and Raspberry Pi. International Journal of Creative Research Thoughts (IJCRT), 6(2), April 2018.
10. N. Dhanasekar, C. Valavan, and S. Soundarya. IoT based intelligent bus monitoring system. International Journal of Engineering Research & Technology (IJERT), 7(11), 2019.
11. M. S. Minu and K. N. Deepak Adithya. Real time college bus monitoring and notification system. International Research Journal of Engineering and Technology (IRJET), 7(4), Sep 2018.
12. A. Ahmed, E. Nada, and W. AlMutiri. University buses routing and tracking system. International Journal of Computer Science & Information Technology (IJCSIT), 9(1), February 2017.
13. S. Phadatare, P. Kudtarkar, J. Seth, and V. Yeole. Bus monitoring system using Android application. International Research Journal of Engineering and Technology (IRJET), 6(4), Apr 2019.
14. M. A. Hannan, A. M. Mustapha, A. Hussain, and H. Basri. Intelligent bus monitoring and management system. In Proceedings of the World Congress on Engineering and Computer Science, Volume II, San Francisco, USA, October 2012.
15. P. Shanmuga Priya, P. Lavanya, K. Preethika, S. Sandhiya, and B. Subhapriya. GPS based tracking system for government buses. International Journal of Engineering Research & Technology (IJERT), 13(5), June 2025.
16. N. Khwaja, S. Kadivala, S. M. Musharaf, and V. Baloda. Real time bus tracking system. International Journal of Advanced Research in Science, Communication and Technology (IJARSCT), 2(3), April 2022.
17. D. Vijay Kumar. GPS and GSM based passenger tracking system. International Journal of Computer Applications, 100(2), August 2014.
18. M. R. S. Madhur, K. S. Priyatham, K. C. Raj, K. B. Nandan, and B. V. Sathish Kumar. Smart bus tracking and management system. Iconic Research and Engineering Journals (IRE), 1(8), Feb 2018.
19. M. Kumbhar, M. Survase, P. Mastud, and A. Salunke. Real time web based bus tracking system. International Research Journal of Engineering and Technology (IRJET), 3(2), Feb 2016.), 3(2), Feb 2016.