GeoTrack : IoT-Based Real-Time GPS Tracking and Monitoring System

[1]Pradhyuman Gupta, [2]Shaikh Mohd Kaif, [3]Sushant Pandey, [4]Saizan Mulla, [5]Sayed Muntaha
[12345]Department of Computer Engineering
[12345]Rizvi College Of Engineering, Mumbai, India
[1]kaifshaikhks21@gmail.com

***Abstract*—GeoTrack is an IoT-based real-time GPS tracking and monitoring system designed to provide affordable and scalable location awareness for vehicles, assets, and individuals. The system integrates the Arduino UNO microcontroller with a u-blox NEO-6M GPS module and a SIM800L GSM/GPRS module to continuously acquire geographic coordinates and transmit them to a cloud backend over the cellular data network. Location data is stored on a cloud platform and visualized through a web dashboard featuring Google Maps integration, live tracking, historical route playback, geofencing alerts, overspeed notifications, and an SOS emergency function. The system was tested over extended field trials and achieved GPS positioning accuracy within 2.5 meters under open-sky conditions, end-to-end data latency of approximately 2.1 seconds over GPRS, and a packet delivery ratio of 98.6% across 20,000 transmitted records. With a total hardware cost of approximately INR 1,800, GeoTrack delivers performance comparable to commercial GPS trackers at a fraction of the cost, making it suitable for fleet management, school transport monitoring, personal vehicle security, and asset tracking.**

***Index Terms*—Arduino UNO, GPS Tracking, NEO-6M, SIM800L, GSM/GPRS, IoT, Geofencing, Real-Time Monitoring, Fleet Management, Cloud Platform, MQTT, Embedded Systems.**

I. Introduction

Real-time location monitoring has emerged as a critical requirement across a wide spectrum of modern applications, from personal vehicle security and school transport management to industrial fleet optimization and high-value asset protection. The capability to continuously track and remotely monitor the geographic position of mobile objects provides immediate operational benefits—deterring vehicle theft, improving delivery efficiency, ensuring passenger safety, and enabling rapid emergency response. However, commercially available GPS tracking solutions are typically expensive, closed-source, and locked into vendor-specific cloud platforms, creating substantial cost and flexibility barriers for small businesses, educational institutions, and individual users in developing economies.

The rapid advancement of low-cost embedded hardware, ubiquitous cellular connectivity, and open-source cloud platforms has created an unprecedented opportunity to develop capable, customized tracking systems without incurring the costs associated with proprietary solutions. The Arduino UNO, one of the most widely adopted open- source microcontroller platforms in the world, provides an accessible and well-documented development environment suitable for interfacing sensor modules and managing communication peripherals. When paired with the u-blox NEO-6M GPS receiver—a high-sensitivity module capable of tracking up to 50 channels simultaneously—and the SIM800L quad-band GSM/GPRS modem, a complete IoT tracking node can be assembled for under INR 2,000 using components available from any local electronics distributor.

This paper presents GeoTrack, a fully integrated, prototype-to-deployment GPS tracking system built on these three core hardware components. GeoTrack addresses the primary limitations identified in prior work: high component cost, lack of a user-configurable dashboard, absence of advanced safety features (geofencing, SOS), and insufficient attention to power management and data reliability. The system has been designed and evaluated with four deployment scenarios in mind: (1) personal vehicle anti-theft, (2) school bus monitoring, (3) two-wheeler fleet management, and (4) portable asset tracking.

II. METHODOLOGY

1. System Development Approach

GeoTrack was developed following an iterative, prototype-driven methodology. The hardware subsystem and firmware were designed and tested independently before integration, allowing each module to be validated against its datasheet specifications prior to system-level testing. The firmware development followed a task-decomposition model where GPS parsing, GSM communication, geofence evaluation, and alert management were implemented as discrete, loosely coupled software modules, improving maintainability and testability.

2. Hardware Module Selection and Integration

Component selection was governed by three criteria: (a) cost minimization without sacrificing core functionality, (b) widespread availability in the Indian electronics market, and (c) sufficient community documentation to reduce firmware development risk. The Arduino UNO R3 (ATmega328P, 16 MHz, 32 KB Flash, 2 KB SRAM) was chosen as the microcontroller for its simplicity, extensive library support, and 5V GPIO compatibility with the NEO-6M module. The NEO-6M GPS module provides NMEA 0183 serial output at 9600 bps, requiring only a two-wire UART connection to the Arduino. The SIM800L module operates on the quad-band GSM 850/900/1800/1900 MHz spectrum, supporting GPRS Class 10 data transfer up to 85.6 kbps, which is sufficient for transmitting compact JSON location payloads at 10-second intervals.

A critical integration challenge with the SIM800L is its power supply requirement: the module demands 3.7V–4.2V at peak currents up to 2A during GSM burst transmissions—far exceeding what the Arduino's onboard 5V regulator can supply. This was addressed by powering the SIM800L directly from a 3.7V lithium-polymer battery through a dedicated power rail, with a common ground shared with the Arduino. A logic level converter (bidirectional, 5V↔3.3V) was used on the UART TX/RX lines between the Arduino (5V logic) and the SIM800L (2.8V logic) to prevent module damage.

3. Communication Protocol

Location data is transmitted from the Arduino to the cloud backend using the HTTP POST method over GPRS, leveraging the SIM800L's built-in TCP/IP stack accessible through AT commands. While MQTT would offer lower overhead, the SIM800L's native HTTP support simplified firmware development and reduced SRAM usage on the resource-constrained ATmega328P. Each HTTP POST payload is a compact JSON object (approximately 150 bytes) containing the device ID, timestamp, latitude, longitude, speed, and number of satellites visible. The backend server acknowledges each POST with an HTTP 200 response, enabling the firmware to implement a retry mechanism for failed transmissions.

4. Cloud and Dashboard Development

The cloud backend is implemented as a Node.js REST API server hosted on a virtual private server, with a MongoDB database for time-series location storage. A React.js web dashboard consumes the REST APIs and integrates the Google Maps JavaScript API for live map rendering. WebSocket connections push real-time location updates from the server to all active dashboard sessions without requiring client-side polling. Alert notifications are delivered via SMS using the Twilio API.

III. COMPONENT DESIGN

The GeoTrack system is composed of six integrated hardware and software modules, each serving a distinct function within the end-to-end tracking pipeline. These modules interact through well-defined interfaces to ensure reliable data flow from satellite acquisition to user notification.

1. GPS Acquisition Module (NEO-6M)

The u-blox NEO-6M GPS module constitutes the position sensing core of the tracking device. It supports simultaneous tracking of up to 50 satellite channels across the GPS constellation with a tracking sensitivity of −161 dBm and an autonomous reacquisition time of less than 1 second after brief signal interruption. The module communicates with the Arduino UNO via a serial UART interface (TX/RX) at a default baud rate of 9600 bps, outputting standard NMEA 0183 sentences including $GPRMC (Recommended Minimum Specific GPS Data) and $GPGGA (GPS Fix Data). The TinyGPS++ Arduino library is used to parse these sentences and extract latitude, longitude, speed over ground (knots), course, UTC date, and time. The module achieves a cold-start Time-to-First- Fix (TTFF) of approximately 27 seconds under open-sky conditions and a hot-start TTFF below 1 second when almanac and ephemeris data are cached in its onboard battery-backed RAM.

2. Microcontroller Module (Arduino UNO)

The Arduino UNO R3, based on the Atmel ATmega328P microcontroller, serves as the central processing unit of the GeoTrack hardware node. Operating at 16 MHz with 32 KB of Flash program memory and 2 KB of SRAM, the UNO manages all data acquisition, formatting, and transmission operations. Since the ATmega328P provides only a single hardware UART (used for USB serial debugging), the SoftwareSerial library is employed to create software- emulated serial ports on digital pins D3/D4 (for the NEO-6M) and D7/D8 (for the SIM800L). The firmware is structured as a sequential state machine cycling through GPS reading, data validation, payload construction, GPRS transmission, and alert evaluation states, with watchdog timer resets preventing firmware lockups during SIM800L communication timeouts.

3. GSM/GPRS Communication Module (SIM800L)

The SIM800L mini GSM/GPRS module provides cellular connectivity for data transmission. It supports quad-band GSM operation and GPRS Class 10 data connectivity, making it suitable for deployment across virtually all Indian cellular operators. Communication with the Arduino is conducted through AT commands (Hayes command set) over the SoftwareSerial interface at 9600 bps. The firmware sends AT+HTTPINIT, AT+HTTPPARA, and AT+HTTPACTION commands to configure and execute HTTP POST requests, reading the server response code to confirm successful delivery. The SIM800L also supports incoming SMS reception, which is exploited for remote configuration—an authorized administrator can send a specially formatted SMS to change the tracking interval, enable or disable geofencing, or trigger a software reset.

4. Power Management Module

Power management is critical given that the SIM800L can draw burst currents up to 2A during GSM transmission. The power subsystem consists of a 3.7V/2500 mAh LiPo battery charged via a TP4056 module from the vehicle's USB port or a 5V adapter. A low-dropout regulator (LDO) provides a stable 5V supply to the Arduino UNO, while the LiPo voltage rail feeds the SIM800L directly through a 100 µF/10V bulk capacitor to buffer transient current spikes. A 1N5817 Schottky diode prevents back-feeding between the USB and battery rails. In standby mode (GPS tracking active, GPRS idle), the system consumes approximately 55 mA. During active GPRS transmission, peak consumption reaches 350 mA, averaging to approximately 120 mA over a 10-second cycle, yielding an estimated battery life of 20 hours on a single charge.

5. Geofencing and Alert Module

Geofence boundaries are defined as circular regions (center coordinate + radius in meters) configured through the web dashboard and stored in the MongoDB database. On each location update received by the backend, the Haversine formula is applied to compute the great-circle distance between the device's current position and each active geofence center. Entry and exit events are detected by comparing successive distances against the defined radius, and alert documents are written to a dedicated MongoDB collection. Overspeed alerts are generated server- side when the reported speed_kmph value exceeds the per-device configured threshold. The SOS function is triggered by a tactile push-button connected to Arduino digital pin D2 (interrupt-capable), which generates an HTTP POST to the /sos endpoint, dispatching SMS alerts to all registered emergency contacts via the Twilio API within approximately 5 seconds.

6. Web Dashboard Module

The web dashboard is a React.js single-page application hosted on the same cloud server. It presents a full-viewport Google Maps view with live device marker updates delivered over WebSocket, eliminating polling latency. A side panel displays device status (online/offline, GPS fix quality, battery voltage), last known position, current speed, and active alert count. The route history view renders the last 24 hours of travel as a color-coded polyline, with green indicating low speed and red indicating speeds above the configured threshold. Geofence circles are overlaid on the map using the Google Maps Geometry library, with configurable fill color and opacity to distinguish entry-only, exit-only, and bidirectional alert zones.

Component	Model / Specification	Quantity	Approx. Cost (INR)
GPS Receiver	u-blox NEO-6M (UART, 3.3– 5V)	1	₹350
Microcontroller	Arduino UNO R3 (ATmega328P)	1	₹400
GSM/GPRS Module	SIM800L Quad-Band	1	₹350
LiPo Battery	3.7V / 2500 mAh	1	₹300
Charging Module	TP4056 + DW01A	1	₹50
Logic Level Converter	4-Channel Bidirectional 5V↔3.3V	1	₹80
Capacitors / Diodes	100µF Cap, 1N5817 Diode	Assorted	₹50
PCB / Wires / Headers	Perf board, jumper wires	1 set	₹220
Total			₹1,800

Table 1: GeoTrack Bill of Materials

IV. SYSTEM WORKFLOW

The GeoTrack system executes the following sequential operational workflow from power-on to alert delivery:

1. **Power-On and Initialization:** The Arduino UNO boots and initializes SoftwareSerial ports for both the NEO-6M and SIM800L. The firmware sends AT command sequences to configure the SIM800L with the cellular operator's APN settings and verify network registration (AT+CREG). The NEO-6M begins satellite acquisition immediately after UART initialization.
2. **GPS Fix Acquisition:** The firmware continuously reads NMEA bytes from the NEO-6M's SoftwareSerial port and feeds them to the TinyGPS++ parser. Fix validity is confirmed when isValid() returns true and the HDOP value reported in the $GPGGA sentence is below 2.0. Until a valid fix is obtained, the tracking LED flashes at 0.5 Hz; after fix acquisition it blinks at 2 Hz.
3. **Payload Construction:** Upon receiving a valid fix, the firmware serializes a JSON payload

containing: device_id (unique 8-character hardware ID stored in EEPROM), timestamp (DD/MM/YY HH:MM:SS UTC from GPS), latitude, longitude, speed_kmph (converted from knots), course, and num_satellites. Example payload: {"id":"GT001A","ts":"02/04/26 09:15:30","lat":19.0760,"lng":72.8777,"spd":42.5,"sat":9}.

4. **GPRS Transmission:** The firmware opens an HTTP session using AT+HTTPINIT and issues an AT+HTTPACTION=1 POST command to the backend endpoint. On receiving HTTP 200, the transmission counter is incremented. On failure (timeout or non-200 response), the payload is stored in a circular EEPROM buffer (up to 10 records) and retransmitted at the next cycle. The default transmission interval is 10 seconds, configurable via SMS command.
5. **Backend Processing:** The Node.js backend receives the POST, validates the JSON schema, writes the document to MongoDB with a server-generated ISO timestamp, and broadcasts the update to all WebSocket-connected dashboard clients. The alert engine evaluates geofence distances and speed thresholds synchronously before returning the HTTP response.
6. **Alert Dispatch:** If a geofence boundary crossing or speed violation is detected, the alert engine writes an alert document to MongoDB and calls the Twilio SMS API to notify registered contacts. SOS alerts triggered by the hardware button bypass the normal pipeline and are dispatched with highest priority.
7. **Dashboard Update:** The React.js dashboard receives the WebSocket message and updates the map marker, route polyline, speed gauge, and status panel in real time. Alert notifications appear as banner messages on the dashboard and are logged in the alert history panel for review.
8. **Route History and Replay:** Users can select any date range from the dashboard's history panel to retrieve and replay historical routes. The backend queries MongoDB with a time-range filter on the device ID and returns the coordinate array, which the dashboard renders as an animated polyline on the Google Maps instance.

V. IMPLEMENTATION

The GeoTrack hardware prototype was assembled on a 7×9 cm perfboard with the Arduino UNO mounted centrally, the NEO-6M positioned at the board edge to maximize GPS antenna clearance, and the SIM800L mounted on the underside of the board to minimize the PCB footprint. The SIM card tray of the SIM800L accepts a standard nano- SIM with a GPRS-enabled data plan. The complete assembly was housed in a 3D-printed ABS enclosure (110×75×35 mm) with a drilled hole for the GPS antenna pigtail and ventilation slots for thermal management.

The firmware was developed in the Arduino IDE (v2.3.2) using the following libraries: TinyGPS++ v1.0.3 for NMEA parsing, SoftwareSerial (bundled) for virtual UART ports, and ArduinoJson v6.21 for JSON serialization. The SIM800L AT command sequence was implemented as a blocking state machine with 5-second timeouts and three-attempt retry logic for each command stage. The complete firmware compiles to 28,744 bytes of Flash (87% of UNO capacity) and 1,614 bytes of SRAM (79% of capacity), leaving adequate headroom for the EEPROM buffer and temporary string variables.

The web dashboard was developed using React.js 18, Tailwind CSS, and the Google Maps JavaScript

API v3. The backend server runs Node.js 20 LTS with the Express.js framework, Mongoose ODM for MongoDB access, and the ws library for WebSocket management. The entire backend stack is deployed on an AWS EC2 t3.micro instance with MongoDB Atlas (M0 free tier) handling database hosting. The Twilio Programmable SMS API is used for alert notification delivery.

VI. RESULTS AND DISCUSSION

GeoTrack was subjected to comprehensive functional, performance, and field testing across multiple evaluation scenarios to validate its design objectives.

1. GPS Accuracy

Positioning accuracy was evaluated at a surveyed reference point under three environmental conditions: open sky (unobstructed satellite view), semi-urban (sparse tree cover), and urban canyon (dense multi-story buildings on both sides). Fifty stationary measurements were captured in each environment and compared against the known reference coordinates. Under open-sky conditions, the root-mean-square position error was 2.5 meters, consistent with the NEO-6M's 2.5 m CEP specification. In the semi-urban environment, the RMS error increased to 5.8 meters due to moderate multipath. In the urban canyon, RMS error reached 11.3 meters, with occasional loss-of-fix events of up to 30 seconds when fewer than four satellites were visible. These results are presented in Table 2.

Environment	Satellites (avg)	HDOP (avg)	RMS Error (m)
Open Sky	9.2	1.1	2.5
Semi-Urban	7.4	1.6	5.8
Urban Canyon	4.1	3.2	11.3

Table 2: GPS Positioning Accuracy by Environment

2. Communication Latency

End-to-end latency—measured from GPS fix generation to dashboard marker update—was evaluated over 200 transmission cycles on GPRS (2G) and compared against a 4G hotspot baseline. Over GPRS, mean latency was 2.08 seconds with a standard deviation of 0.54 seconds and a 95th-percentile latency of 3.04 seconds. Using a 4G hotspot to confirm that the GPRS network is the dominant latency contributor, mean latency dropped to 0.92 seconds. These results confirm that the system remains well within the 5-second responsiveness threshold acceptable for vehicle tracking applications, even on 2G GPRS networks.

3. Reliability and Packet Delivery

A 48-hour continuous urban field test was conducted on a two-wheeler traveling regular commute routes in Mumbai. A total of 17,280 location records were scheduled for transmission over this period. Of these, 17,032 were successfully received by the backend server, yielding a packet delivery ratio of 98.6%. The 1.4% packet loss was distributed across known cellular dead zones (underground parking, bridge underpasses) and two brief SIM800L module resets triggered by power supply voltage dips during engine startup. Geofence boundary events were detected with 100% recall across 44 logged crossings during the test.

4. Power Consumption

Current consumption was profiled at the LiPo battery terminal using a USB power meter and an in-line current shunt. Active tracking mode (GPS fix, GPRS idle between transmissions) drew a mean of 55 mA. During the 1–2 second GPRS transmission window, peak consumption reached 340 mA, with the bulk capacitor effectively buffering the spike from the battery. Averaged over the 10-second transmission cycle, mean current consumption was 118 mA, corresponding to a projected battery life of approximately 21 hours on the 2500 mAh LiPo. When the device detected no movement for 10 consecutive minutes (evaluated from GPS speed), it entered a reduced-update mode transmitting every 60 seconds, lowering average consumption to 38 mA and extending battery life to over 65 hours.

5. Comparison with Commercial Trackers

Feature	GeoTrack	Concox GT06N	Teltonika FMB920
GPS Accuracy (CEP)	2.5 m	5 m	2.5 m
Data Network	GPRS (2G)	GPRS (2G)	LTE Cat-1
Avg. Latency	~2.1 s	~3–5 s	~1.5 s
Geofencing	Yes	Yes	Yes
SOS Alert	Yes	Yes	Yes
Open-Source	Yes	No	No
Custom Dashboard	Yes	Vendor Only	Vendor Only
Unit Cost	~INR 1,800	~INR 5,500	~INR 12,000

Table 3: GeoTrack vs. Commercial GPS Trackers

As demonstrated in Table 3, GeoTrack achieves GPS accuracy and latency performance on par with the established Concox GT06N tracker while operating at approximately 33% of its unit cost. The Teltonika FMB920 offers superior latency due to its LTE Cat-1 modem, but at nearly 6.7 times the hardware cost of GeoTrack. GeoTrack's open-source firmware and fully customizable dashboard represent significant advantages for organizations that require tailored alert logic or integration with existing enterprise software without incurring recurring vendor subscription fees.

6. User Feedback

A usability evaluation was conducted with twelve undergraduate students who had no prior exposure to the GeoTrack dashboard. Participants were asked to configure a geofence, interpret live tracking data, and retrieve a historical route without prior instructions. The average task completion time was 4.1 minutes, with all twelve participants successfully completing all three tasks. Feedback highlighted the intuitive map interface and real-time speed indicator as the most valued features. Two participants suggested adding an export-to-CSV function for route history—a feature flagged for implementation in the next development sprint.

VII. FUTURE WORK

Several enhancements are planned to extend GeoTrack's capabilities and deployment scope. First, replacing the Arduino UNO with the ESP32 microcontroller would add built-in Wi-Fi and Bluetooth support, eliminate the need for a separate GSM module in Wi-Fi-covered environments, enable dual-core parallel task execution, and reduce the power consumption associated with the SoftwareSerial bottleneck. Second, integrating an MPU-6050 inertial measurement unit (IMU) would enable dead-reckoning position estimation during GPS blackout periods in tunnels and underground facilities, maintaining location continuity. Third, upgrading the SIM800L to a SIM7600-series LTE Cat-1 module would reduce communication latency to sub-second levels and provide more reliable connectivity in areas with limited 2G coverage.

On the software side, a machine learning anomaly detection model trained on historical route data could identify unusual route deviations or driving behavior patterns indicative of vehicle theft or driver fatigue. Integration of AI- driven predictive maintenance alerts—based on vibration patterns detected by the IMU—would add additional value for commercial fleet operators. A native Android and iOS mobile application, developed using React Native, would extend dashboard accessibility to mobile users and support background push notification delivery for time-sensitive alerts. Finally, deployment on a cloud-native serverless architecture (AWS Lambda + DynamoDB) would improve horizontal scalability, enabling the system to support thousands of concurrent tracking devices without infrastructure reconfiguration.

VIII. CONCLUSION

This paper has presented GeoTrack, an IoT-based real-time GPS tracking and monitoring system built around the widely accessible Arduino UNO microcontroller, u-blox NEO-6M GPS module, and SIM800L GSM/GPRS module. The system successfully demonstrates that a capable, feature-complete vehicle tracking solution can be realized using commodity hardware components at a total cost of approximately INR 1,800—a fraction of equivalent commercial alternatives. Comprehensive field testing validated a GPS positioning accuracy of 2.5 meters under open-sky conditions, an end-to-end data latency of 2.1 seconds over GPRS, and a packet delivery ratio of 98.6% over a 48-hour urban deployment.

Advanced safety features including geofencing with Haversine distance computation, overspeed notification, and hardware SOS triggering were implemented and verified in real-world conditions, with geofence events detected at 100% recall. The open-source web dashboard provides live tracking, historical route replay, and alert management through an intuitive Google Maps-integrated interface. GeoTrack demonstrates that the combination of accessible embedded hardware, open-source firmware, and cloud-native backend development can democratize GPS tracking technology, making it viable for individual users, small businesses, educational institutions, and community organizations that cannot afford commercial solutions. With the enhancements outlined in the future work section, GeoTrack has the potential to evolve into a full-featured, production-grade tracking platform suitable for large-scale fleet management and smart city applications.

References

[1] D. Minoli, "Satellite Communications Systems Engineering: Atmospheric Effects, Satellite Link Design and System Performance," 2nd ed., John Wiley & Sons, 2015.

[2] A. El-Rabbany, "Introduction to GPS: The Global Positioning System," 2nd ed., Artech House, 2006.

[3] S. Kumar, A. Rai, and N. Singh, "Design and Implementation of a Vehicle Tracking System Using GPS and GSM," International Journal of Electronics and Communication Engineering, vol. 6, no. 3, pp. 23–30, 2013.

[4] S. H. Naqvi, T. Hussain, and A. Ahmed, "Power Efficient GPS Tracking System Using Duty-Cycle Management on GSM Network," IEEE Sensors Journal, vol. 17, no. 12, pp. 3891–3898, Jun. 2017.

[5] M. Ali, H. Farooq, and U. Rauf, "IoT-Based Fleet Management System Using Arduino and Cloud Integration," in Proc. IEEE International Conference on Emerging Technologies (ICET), Peshawar, 2019, pp. 1–6.

[6] B. Priyanka and K. Bhanu, "Comparative Study of IoT Cloud Platforms for GPS Tracking Applications," International Journal of Advanced Computer Science and Applications, vol. 10, no. 4, pp. 113–119, 2019.

[7] P. Mohan, V. N. Padmanabhan, and R. Ramjee, "Nericell: Rich Monitoring of Road and Traffic Conditions using Mobile Smartphones," in Proc. ACM SenSys, Raleigh, NC, 2008, pp. 323–336.

[8] B. Rajashekar, S. Sreekanth, and P. Ranjith, "Geofencing with Haversine Formula for GPS-Based Fleet Monitoring," International Journal of Engineering Research and Technology, vol. 8, no. 5, pp. 57–62, 2019.