WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

SafeCircle AI: A transparent rule-based route safety assessment and emergency support web application for women travellers in Tamil Nadu

Mrs. R.Jenifer[1] , Sriram D[2], Srinithi B[3], Srinithi RJ[4], Srimathi P.[5]

Department of Computer Science and Engineering, Sri Shakthi Institute of Engineering and Technology , Coimbatore, Tamilnadu, India

DOI : https://doi.org/10.55248/gengpi.07.1026.34080

A B S T R A C T :

Concern for personal safety shapes how, when and where many women travel, yet mainstream navigation services rank routes almost entirely by travel time or distance. This paper presents SafeCircle AI, a full-stack web application restricted to Tamil Nadu, India, that adds a transparent safety assessment and emergency assistance to conventional routing. A React and Leaflet client and a Node.js/Express server obtain alternative routes from Open Source Routing Machine (OSRM) services and score sampled points along each route with deterministic rules that combine OpenStreetMap features from the Overpass API (emergency facilities, unlit roads and roads under construction), current Open-Meteo weather, community reports and a night-time factor. The scores are rule-based, not learned; the AI component is a multilingual assistant that answers routine questions from templates built on the computed context and forwards other questions to a Google Gemini model, with speech input and output in six Indian languages. Emergency support combines a hold-to-activate SOS with a cancellable countdown, alerts carrying the user's location to trusted contacts, one-tap calls to contacts, the national emergency number 112 and district police control rooms, and time-limited links for sharing a live journey. The platform also provides authentication, administrator moderation of community reports and offline support. A code-level review, a production build, the bundled unit tests and deterministic re-execution of the scoring functions confirm that the routing and scoring pipeline works and identify its main limitations, notably a heuristic traffic estimate and a rank-based adjustment that overrides computed route scores. No user study or field validation has yet been conducted. The paper documents the design and implementation and sets out the corrections required before an empirical evaluation.

Keywords: Women's safety; Safety-aware routing; OpenStreetMap; Rule-based risk scoring; Emergency SOS; Large language model assistant; Web GIS; Tamil Nadu

Introduction

Personal safety strongly influences how women use streets and public transport. Perceived risk of harassment has measurable consequences: in Delhi, women have been found willing to choose a lower-quality college in exchange for a travel route perceived as safer (Borker, 2021), and research on transit environments shows that women's fear is concentrated in isolated, poorly lit and poorly supervised settings (Loukaitou-Sideris, 2014). Mainstream navigation services, by contrast, rank routes by travel time or distance over a road graph (Dijkstra, 1959; Geisberger et al., 2008; Luxen & Vetter, 2011) and leave every judgement about safety to the traveller.

Research on safety-aware routing has shown that routes can be ranked by other criteria when suitable data exist. Risk estimates derived from recorded crime (Galbrun et al., 2016), crowd-sourced incident reports (Shah et al., 2011), the sentiment of geotagged social-media posts (Kim et al., 2014) and combinations of official and crowd-sensed data (Mata et al., 2016) have all been used to recommend safer paths, and perceptual qualities of streets have been operationalised for route recommendation (Quercia et al., 2014). In parallel, personal-safety smartphone applications have multiplied, although reviews report little evidence of their effectiveness and question whether they make users safer or more vulnerable (Eisenhut et al., 2020; Maxwell et al., 2020).

Two practical constraints shape such a system in India. First, the country operates a single emergency number, 112, under the Emergency Response Support System (ERSS) (Ministry of Home Affairs [MHA], n.d.), so emergency features should lead users to this service rather than to proprietary channels. Second, official crime statistics are published annually in aggregated form (National Crime Records Bureau [NCRB], 2026), but, to the authors' knowledge, no public street-level, near-real-time incident feed exists for Tamil Nadu. A route-safety tool for the state must therefore rely on proxies — mapped infrastructure, environmental conditions and community reports — and must be explicit about what those proxies can and cannot show.

This paper describes SafeCircle AI, a web application developed as an undergraduate project for this setting. The application is restricted to Tamil Nadu by a polygon geofence. It retrieves alternative routes from OSRM, scores each route with a deterministic rule set that combines OpenStreetMap infrastructure tags, distance to emergency facilities, current weather, community reports and time of day, and records the factors behind each score. It

also offers SOS alerting to trusted contacts, one-tap emergency calling and messaging, a community hazard-reporting map with administrator moderation, live GPS tracking and journey sharing, offline support, and a multilingual assistant that uses a large language model (LLM) for open-ended questions. The contributions of the paper are:

- a full-stack reference implementation that integrates open geocoding, routing, mapping, facility and weather services for one Indian state, with a fallback for each service;
- a transparent, rule-based location and route scoring scheme, specified completely in Section 3 so that its behaviour can be inspected and criticised;
- a data-provenance design in which every observation carries its source type, verification status and a confidence value, and in which synthetic demonstration incidents are excluded from scoring; and
- a code-level review of the prototype that documents the implementation of each feature and the corrections needed before an empirical evaluation.

Because SafeCircle AI is a prototype, the paper deliberately reports what the source code does rather than what the interface suggests. No user study, field validation, accuracy measurement or load test has yet been carried out, and none is reported. Section 2 reviews related work, Section 3 describes the design and scoring models, Section 4 the implementation and Section 5 the verification performed. Section 6 discusses the findings, and Sections 7–9 present limitations, future work and conclusions.

1.1. Problem statement

Given an origin and a destination inside Tamil Nadu and a travel mode (walking, cycling or driving), the system should (i) return a small set of road-network routes; (ii) attach to each route an interpretable indicator of relative exposure to known risk-related conditions, derived only from openly available or user-contributed data and accompanied by a statement of data coverage; and (iii) give the traveller fast access to trusted contacts and emergency services during the trip — without claiming that any route is safe.

1.2. Objectives

The objectives of the project were to:

- integrate open geocoding, routing, map, facility and weather services into one web application restricted to Tamil Nadu;
- design a rule-based scoring scheme for locations and routes whose factors and parameters are explicit;
- label the provenance and confidence of every observation and keep demonstration data out of live results;
- provide emergency assistance through device-native telephone and messaging channels to trusted contacts, the national number 112 and district police control rooms;
- provide a text and voice assistant in Tamil, English, Hindi, Telugu, Malayalam and Kannada that answers from the computed context; and
- document the implementation and its limitations accurately.

Literature review

1.3. Safety-aware route recommendation

Classical route planning minimises a single additive edge cost such as length or travel time. Dijkstra's algorithm solves this problem exactly (Dijkstra, 1959), and preprocessing techniques such as contraction hierarchies make large-scale queries fast enough for interactive services (Geisberger et al., 2008); OSRM applies these ideas to OpenStreetMap data (Luxen & Vetter, 2011). Safety-aware routing changes the cost function or adds a second objective. Galbrun et al. (2016) estimate a crime-risk value for road segments from historical incident records and compute paths that trade off length against accumulated risk. CROWDSAFE collects crime incidents through crowd-sourcing and uses them for safe routing on mobile devices (Shah et al., 2011). SocRoutes treats the sentiment of geotagged tweets as a safety proxy when choosing routes (Kim et al., 2014), and Mata et al. (2016) combine crowd-sensed reports with official crime data to find safe routes in Mexico City. More broadly, Quercia et al. (2014) used crowd-sourced perceptions of streets to recommend routes perceived as more beautiful, quieter or happier than the shortest path.

These systems depend on point-level incident data, usually recorded crime. Where such data are unavailable, as for street-level incidents in Tamil Nadu, a system can rely only on proxies, and its output should be described as an indicator of exposure to the proxied conditions rather than as a prediction of crime.

1.4. Built environment, lighting and perceived safety

Crime prevention through environmental design (CPTED) holds that features such as lighting, sight lines and the presence of other people ("natural surveillance") influence both offending and fear of crime (Cozens et al., 2005). Evidence on lighting is the strongest: a Campbell systematic review found that improved street lighting was associated with reductions in crime (Welsh & Farrington, 2008), and a randomised experiment in New York City public housing developments found that additional lighting reduced outdoor night-time crime (Chalfin et al., 2022). These findings motivate the use of unlit roads, night-time and distance from staffed facilities as risk-related factors. They do not, however, supply calibrated weights: the numerical parameters used by SafeCircle AI (Section 3) are design choices, not estimates derived from this literature.

1.5. Personal-safety applications and emergency response

A systematic review of mobile applications addressing violence against women found that most focus on emergency and prevention functions and that few have been evaluated (Eisenhut et al., 2020). A content analysis of personal-safety applications questioned whether such applications keep users safe or make them more vulnerable (Maxwell et al., 2020). In India, ERSS integrates police, fire and ambulance response behind the number 112 and offers its own mobile application (MHA, n.d.). In Tamil Nadu, the state police launched the Kavalan SOS application in 2018; pressing its SOS button shares the user's GPS location with the police (Shekar, 2019). Applications of this kind are designed to summon help; few combine emergency features with a pre-trip assessment of route conditions, which is the combination explored here.

1.6. Open geospatial data and services

OpenStreetMap (OSM) is a volunteered geographic information (VGI) project whose data are contributed by individuals rather than by surveying agencies (Goodchild, 2007; Haklay & Weber, 2008). Its coverage of attributes such as street lighting (the lit tag) is uneven, so the absence of a tag cannot be read as the absence of a feature. OSM data can be queried by area through the Overpass API (OpenStreetMap Wiki contributors, n.d.), geocoded through Nominatim — whose public instance is subject to a usage policy that limits request rates (OpenStreetMap Foundation Operations Working Group, n.d.) — and routed with OSRM. Open-Meteo provides current weather variables, including World Meteorological Organization (WMO) weather codes, without an API key (Zippenfenig, 2023). Browser standards such as the Geolocation API (Cáceres & Grant, 2026) and the Web Speech API (Liu, 2026) allow position tracking and speech interfaces inside a web page.

1.7. Large language model assistants

LLMs such as Google's Gemini family (Gemini Team, 2023) answer open-ended questions in many languages, but they can produce fluent statements that are not supported by their inputs, commonly called hallucinations (Ji et al., 2023). Supplying relevant retrieved context with the prompt is a widely used mitigation (Lewis et al., 2020). In a safety application the risk is acute: a fabricated police-station name or distance could mislead a person in distress. This motivates restricting the LLM to questions that cannot be answered from structured data and ensuring that every fact placed in its context is genuine (Section 6.3).

1.8. Research gap

Published safe-routing systems assume access to point-level crime data and have mostly been demonstrated in cities where such data are open, while safety applications concentrate on alerting. Little published work describes a system for an Indian state that (a) uses open infrastructure, weather and community data as transparent proxies in the absence of street-level crime data, (b) integrates them with device-native emergency channels and a multilingual assistant, and (c) reports its own limitations at code level. SafeCircle AI is a first step in this direction; its scoring is heuristic and unvalidated, and this paper treats it as such.

System design and methodology

1.9. Design principles

The design follows five principles, which are stated in the project specification and are partly realised in the code:

- *Indicator, not guarantee.* Scores are meant to be worded as assessments of available data (for example, "Lower Assessed Risk (Available Data)") rather than as statements that a route is safe.
- *Provenance.* Every observation carries a source type (osm_overpass, open_meteo, community_report or demo_simulation), a verification status and a confidence value between 0 and 100.
- *No synthetic crime data in live mode.* Three seeded incident records are flagged as demonstration data. They are never used in scoring and appear on the map only when the user enables Demonstration Mode, with their titles prefixed "[DEMO SIMULATION]". District-level statistical context is shown only as text and is never converted into map points.
- *Regional restriction.* All coordinates are validated against a Tamil Nadu boundary polygon (Section 3.3).
- *Graceful degradation.* Each external service has a timeout and a fallback, so that the application remains usable when a service is unavailable.

Section 6 discusses where the evaluated build departs from these principles.

1.10. System architecture

Fig. 1 shows the three-tier architecture. The browser client is a React single-page application with views for the landing page, sign-in, dashboard, route planner, "Threat Radar" map, hazard reporting, assistant, trusted contacts, journey sharing and administration. It renders maps with Leaflet and uses browser interfaces for geolocation, speech, audio recording, battery and network status. The application server is a single Node.js process running Express. It exposes a JSON REST API, makes all calls to external services, computes the scores and holds the data store. During development the same

process serves the client through Vite middleware; in production it serves the static build. Because external calls are made by the server, the Gemini API key is read from a server-side environment variable and never reaches the browser. Table 1 summarises the technology stack.

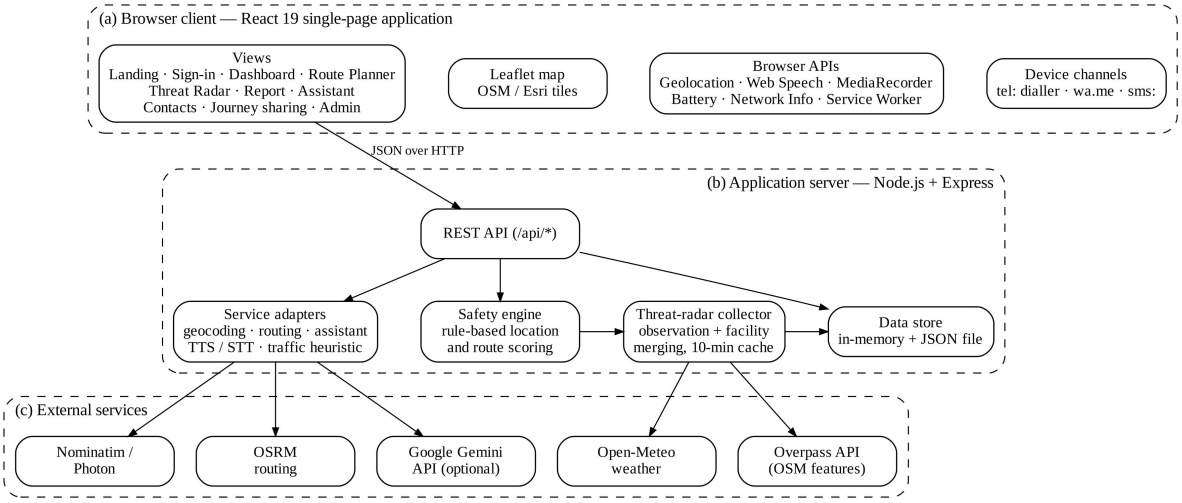


Fig. 1 - System architecture of SafeCircle AI as implemented in the evaluated build.

Table 1 - Technology stack of the evaluated build.

Layer	Technology	Role in SafeCircle AI
Client interface	React 19, React Router 7, TypeScript, Tailwind CSS 4, Lucide icons	Single-page application for planning, reporting, assistance, journey sharing and administration
Mapping	Leaflet 1.9; OpenStreetMap and Esri World Imagery tiles	Route polylines, facility and observation markers, severity circles
Build tooling	Vite 8; tsx	Bundling, development server
Application server	Node.js, Express 4	REST API, orchestration of external services, scoring
Persistence	In-memory store with a JSON file; PostgreSQL via FastAPI	Trusted contacts, saved routes, reports, journeys
Geocoding	Local gazetteer (110 places), Photon, Nominatim	Forward and reverse geocoding restricted to Tamil Nadu
Routing	OSRM instances at routing.openstreetmap.de; OSRM demonstration server as fallback	Three candidate routes per request (padded when OSRM returns fewer)
Context data	Overpass API; Open-Meteo	Facilities, unlit and under-construction roads, weather
Assistant and speech	@google/genai SDK (Gemini); Web Speech API; MediaRecorder	Open-ended answers, speech recognition and synthesis
Python backend	FastAPI, SQLAlchemy, Pydantic, PostgreSQL	Relational data layer for seven entities

1.11. Geographic scope and district resolution

Coverage is restricted to Tamil Nadu. A coordinate is accepted only if it passes a bounding-box pre-check (8.07–13.57° N, 76.21–80.35° E) and an even–odd ray-casting point-in-polygon test (Hormann & Agathos, 2001) against a polygon of 39 distinct vertices that approximates the state boundary. The route-analysis, safety-score, report and radar endpoints return an explicit out-of-region message for any coordinate that fails the test, and geocoder results outside the polygon are discarded. Because the polygon is coarse, it also accepts points in neighbouring territories — for example Puducherry and Karaikal (Union Territory of Puducherry) and Kuppam (Andhra Pradesh) — and may reject points close to the true border.

Each accepted coordinate is assigned to a district by choosing the nearest of 16 district reference points (planar distance in degrees). Each entry stores the district headquarters, the police commissionerate, a police control-room telephone number and a short descriptive summary. Tamil Nadu has 38 districts, so locations in the 22 districts without an entry are attributed to a neighbouring district; district labels are therefore approximate.

1.12. Data sources and observation model

Table 2 lists the data used for scoring. For a query point, the threat-radar collector issues one Overpass query that returns police, hospital, clinic and pharmacy nodes and police and hospital ways within the query radius (clamped to 2–12 km), ways tagged lit=no whose road class is primary, secondary, tertiary or residential within 5 km (footways, service roads and other classes are not queried), and ways tagged highway=construction within 6 km (at most 85 elements in total), together with one Open-Meteo request for current temperature, precipitation, weather code, wind speed and day/night status. Three Overpass mirrors are tried in turn with a 9 s timeout each, and the combined response is cached in memory for 10 minutes, keyed by the coordinates rounded to two decimal places and by the radius.

Overpass results are converted into two record types. Facilities (clinics are classified as hospitals) are merged with a directory of 19 facilities — 8 police stations, 10 hospitals and 1 pharmacy in seven districts — that was compiled manually by the developers; an OSM facility within 120 m of a directory entry of the same type is treated as a duplicate. Unlit and under-construction ways become observations of type *lighting* (severity 3) and *infrastructure* (severity 2), located at the centre point of the way. Weather produces at most one hazard observation: thunderstorm (WMO code ≥ 95, severity 4), rain (precipitation above 2 mm or codes 61–82, severity 3) or fog (codes 45–48, severity 3). The collector then adds directory facilities within 2.5 times the query radius, stored community reports within three times the radius and, when Demonstration Mode is requested, demonstration incidents within 3.5 times the radius; with the default 8 km radius, reports up to 24 km away are therefore included. Each observation records its source type, verification status and a fixed confidence value (for example 86 for unlit roads and 68 for community reports).

Table 2 - Data sources used by the safety engine.

Source	Content	Access and refresh	Use in scoring
OSM via Overpass API	Police, hospital, clinic and pharmacy nodes; police and hospital ways; lit=no on primary to residential roads; highway=construction	Live query per area; 10-min cache; three mirrors	Facility distance (route model); lighting (s = 3) and infrastructure (s = 2) observations
Manually compiled facility directory	19 entries (8 police stations, 10 hospitals, 1 pharmacy) in 7 districts	Static; loaded at start-up	Isolation penalty (location model); merged facility set (route model)
Open-Meteo	Current precipitation, WMO weather code, wind speed, temperature, day/night	Live query; 10-min cache	At most one weather-hazard observation (s = 3–4)
Community reports	Category, severity 1–5, description and location submitted by users	JSON file; stored as "pending" but used immediately	Report term (location model); observations (route model)
Demonstration incidents	Three synthetic incidents in Coimbatore and Chennai	Static; flagged isDemo	Never used in scoring; shown on the map only in Demonstration Mode
District registry	16 districts: reference point, police control number, summary text	Static	District label, explanation text, SOS control-room number

1.13. Location risk model

The location model is used by the dashboard, the assistant and the /api/safety/score endpoint. For a point p at time t it computes a risk value $R \in [0, 100]$ and a safety indicator $S = \max(5, 100 - R)$. Distances are great-circle distances computed with the haversine formula (Sinnott, 1984):

$$d(p,q) = 2R_E \cdot atan2\left(\sqrt{a}, \sqrt{1-a}\right) \tag{1}$$

$$a = sin^2\left(\frac{\Delta\varphi}{2}\right) + cos\,\varphi_p\, cos\,\varphi_q\, sin^2\left(\frac{\Delta\lambda}{2}\right) \tag{2}$$

where φ and λ denote latitude and longitude in radians and R_E = 6 371 km. Three additive terms are then computed. The incident term sums over non-demonstration incidents within 2 km, weighting each by its severity $s \in \{2, 3, 4, 5\}$ (low to critical), a linear distance decay and a recency factor based on its age a in days:

$$I(p) = \sum_{i:d_i \le 2000} 6\, s_i \left(1 - \frac{d_i}{2000}\right) max\left(0.2, 1 - \frac{a_i}{30}\right) \tag{3}$$

The community-report term sums over stored reports within 1.5 km:

$$C(p) = \sum_{j:d_j \le 1500} 4\, s_j \left(1 - \frac{d_j}{1500}\right) \tag{4}$$

The isolation penalty depends on the distance d_P to the nearest police station in the manually compiled directory (OSM police stations are not used here):

$$P(p) = \begin{cases} 0, & d_P \le 2000\ m \\ 6, & 2000\ m < d_P \le 4000\ m \\ 12, & d_P > 4000\ m \\ 10, & \textit{no police station in the directory} \end{cases} \tag{5}$$

Finally, a time-of-day multiplier based on the server clock hour h is applied:

$$\tau(h) = \begin{cases} 1.35, & h \ge 22\ or\ h \le 4 \\ 1.15, & 19 \le h \le 21\ or\ 5 \le h \le 6 \\ 1.00, & otherwise \end{cases} \tag{6}$$

$$R(p,t) = min\{100, round(\tau(h)[I(p) + C(p) + P(p)])\}, \quad S(p,t) = max\{5, 100 - R(p,t)\} \tag{7}$$

S is mapped to four levels: LOW ($S \ge 82$), MODERATE ($65 \le S < 82$), HIGH ($45 \le S < 65$) and CRITICAL ($S < 45$). A confidence of 88 is reported if any directory police station or hospital lies within 10 km, and 65 otherwise. In the current data the incident table contains only demonstration records, which Eq. (3) excludes, so $I(p)$ is always zero: the location score depends only on community reports, distance to a directory police station and time of day (Table 5).

1.14. Route safety model

Candidate routes are requested from the OSRM instances at routing.openstreetmap.de (6.5 s timeout), with the OSRM demonstration server as a fallback (no timeout). The client always receives three candidates: when OSRM returns fewer, the adapter requests routes through perpendicular via-points (4.5 s timeout) and, if these fail or duplicate an existing route, adds synthetic arc geometries whose distances are set to 1.08 and 1.16 times that of the primary route; if routing fails entirely, three synthetic arcs are returned. Such arcs do not follow roads.

For each candidate route the engine queries the threat radar once, at the middle vertex of the route geometry with an 8 km radius, and evaluates a sub-sample of the route's N vertices with step $k = \max(1, \lfloor N/24 \rfloor)$ (Fig. 2). At each evaluated vertex v the penalty π_v and bonus β_v are

$$\pi_v = b(h) + \sum_{o \in W,\, d \le 10\ km} 3\, s_o + \sum_{o \notin W,\, d \le 450\ m} 4.5\, s_o + 5 \cdot \mathbb{1}_{[\delta_v > 3500\ m]}, \quad \beta_v = 6 \cdot \mathbb{1}_{[\delta_v \le 800\ m]} \tag{8}$$

where $b(h) = 8$ between 19:00 and 05:59 (server time) and 2 otherwise, W is the set of weather-hazard observations, $s_o \in [1, 5]$ is the severity of observation o, and δ_v is the distance from v to the nearest facility of any type in the merged set (police station, hospital, clinic or pharmacy). The vertex score is bounded to [25, 98],

$$\sigma_v = min\{98, max\{25, round(100 - \pi_v + \beta_v)\}\} \tag{9}$$

and the route indicator is the rounded mean over the K evaluated vertices:

$$S_r = round\left(\frac{1}{K}\sum_{v=1}^{K} \sigma_v\right) \tag{10}$$

Routes are labelled LOW ($S_r \ge 80$), MODERATE ($55 \le S_r < 80$) or HIGH ($S_r < 55$). The engine also returns "risk zones" (vertices with $\sigma_v < 75$ and at least one contributing observation, with up to two reasons), a list of explanation strings (districts, number of nearby facilities, numbers of unlit-road and community-report encounters, current weather and the night-time factor) and a coverage flag: if neither facilities nor observations are found, the route is labelled "Limited Data Coverage" and its confidence falls from 88 to 60.

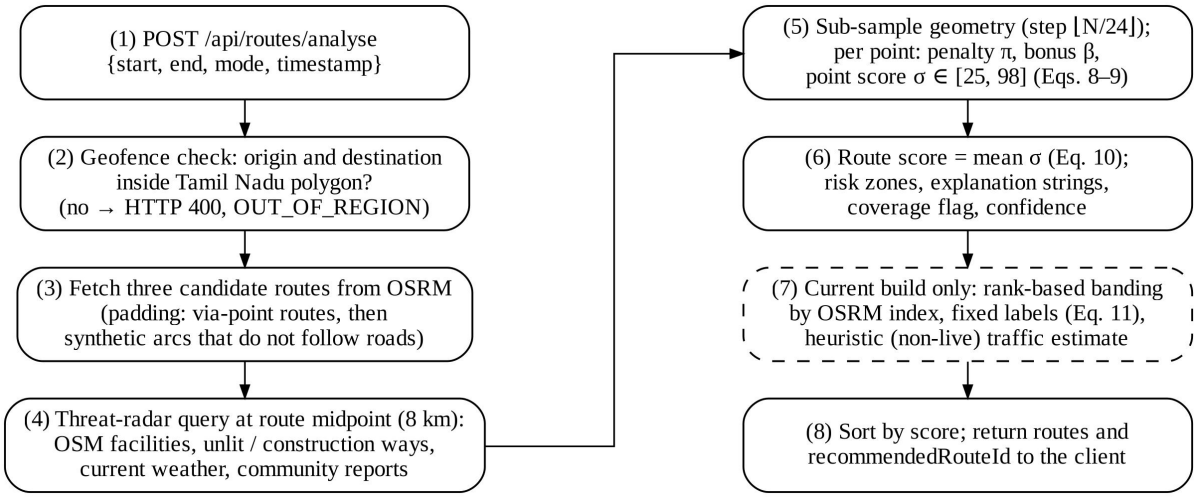


Fig. 2 - Route-analysis pipeline of the /api/routes/analyse endpoint. The dashed step is post-processing present in the evaluated build (Section 3.7).

1.15. Rank-based post-processing in the evaluated build

After S_r has been computed, the route-analysis endpoint applies a further step that is not part of the model above. For the alternative with index k in the order returned by OSRM, the displayed score is

$$S'_r = \begin{cases} min(96, max(88, S_r + 4)), & k = 0 \\ min(87, max(78, S_r - 4)), & k = 1 \\ min(79, max(68, S_r - 10)), & k = 2 \end{cases} \tag{11}$$

and fixed labels are assigned by index ("Safest Recommended Corridor (Well-Lit)", "Fastest Direct Route (Moderate Lighting)" and "Alternative Bypass Route (Lower Patrol Density)"). The routes are then sorted by S'_r and the first is returned as the recommendation. Because the three bands overlap only at 78–79, the recommended route is in practice always OSRM's first alternative, and the labels are not derived from lighting or patrol data. The routing adapter likewise names the candidates "Well-Lit Main Arterial Corridor", "Fastest Direct Express Route" and "Secondary Bypass Corridor" by index, and the first candidate's traffic summary refers to "High CCTV & patrol presence". The displayed risk level is recomputed from S'_r with a third set of thresholds (HIGH < 60, MODERATE < 82), and the "Limited Data Coverage" label of Section 3.6 is overwritten. Because a computed score of 0 — returned when any vertex lies outside the polygon — is replaced by a default of 88 before banding, such a route would still be shown with a score of 92 and the "Safest" label. A comment in the code describes the step as a way to differentiate routes whose computed scores are

identical. It is reported here because it determines what users see; Section 6 discusses its consequences, and its removal is the first correction listed in Section 8.

1.16. Traffic estimate

Each route also carries a congestion estimate, but no traffic data source is queried. The estimate is a deterministic function of the Indian Standard Time hour, the day of the week (taken in UTC), whether the district is one of seven metropolitan districts, proximity to gazetteer landmarks, an optional weather keyword, and a sinusoidal "micro-variance" of ±4 percentage points computed from the coordinates. Per-route values are then offset by fixed amounts (+4, −6 and −14 percentage points for the three alternatives). The interface and the assistant prompt describe this value as "live traffic"; it is a heuristic simulation and is treated as such in this paper.

Implementation

1.17. Client application

The client is written in TypeScript with React 19, React Router 7 and Tailwind CSS 4 and is bundled by Vite. The route planner provides location autocompletion (local gazetteer, then Photon and Nominatim through the server), selection of origin and destination by map click, a choice of walking, cycling or driving, and cards for the three candidates showing the score, label, distance, duration and traffic estimate. Routes are drawn as Leaflet polylines coloured by option and selection; the selected route has a translucent halo coloured by its (banded) risk level. Durations are recomputed on the server from distance using fixed speeds of 4.8, 15.5 and 32 km/h (minimum 60 s) rather than taken from OSRM. Frequently used routes can be saved and reloaded from the planner.

The Threat Radar view lists observations and facilities around a selected centre, with filtering by category and a Demonstration Mode switch. Observations are drawn as circles whose radius is proportional to severity ($76 \cdot s$ metres), which is a weighted point display rather than a kernel-density heat map. The report view lets a user place a pin, choose a category (lighting, road construction, harassment, suspicious activity, theft or other), set a severity from 1 to 5 and submit anonymously. On the server, suspicious activity, theft and other are stored under a single "isolated area" category, and severities 1 and 2 are both stored as "low" (scored as 2). Further views provide sign-in and registration, management of trusted contacts, journey sharing and an administrator dashboard (Sections 4.2, 4.4, 4.5 and 4.7).

1.18. Server application and API

The Express server exposes the endpoints in Table 3. Request bodies are JSON (up to 15 MB, to accommodate recorded audio for transcription). Users register and sign in through token-based authentication, and administrator functions are restricted to accounts with the administrator role. Input validation consists of coordinate parsing, the geofence check and required-field checks; rate limiting is not implemented. The Gemini API key is read from the process environment (the server does not load a .env file), and in production mode the server serves the static build without a fallback route for client-side paths, so deep links require such a route to be added.

Table 3 - Principal REST endpoints of the Node.js server.

Method and path	Function	External dependencies
POST /api/routes/analyse	Alternative routes with scores, labels and traffic estimate	OSRM, Overpass, Open-Meteo
POST /api/safety/score	Location risk (Eqs. 1–7)	None
GET /api/radar/live	Observations, facilities, weather, source status, district context	Overpass, Open-Meteo
GET /api/safety/incidents, /facilities, /heatmap, /sources	Views of radar data with store fallback	Overpass, Open-Meteo
GET /api/location/search; POST /api/location/ReverseGeocode	Forward and reverse geocoding	Photon, Nominatim
GET, POST /api/reports	List and submit community reports	Nominatim (area name)
POST /api/sos/trigger	Raise an SOS alert and notify trusted contacts	Nominatim
GET /api/user/profile; POST, DELETE /api/user/contacts	User profile; trusted contacts	None
Authentication, journey and administration endpoints	Sign-in and registration; journey sessions and share links; report moderation, analytics and audit log	None
POST /api/assistant/query	Assistant answer and structured context	Nominatim, Overpass, Open-Meteo, Gemini
POST /api/tts/speak; POST /api/stt/transcribe	Speech synthesis and transcription	Google Translate TTS endpoint, Gemini

1.19. Data persistence

The Node.js server holds its data in an in-memory store initialised from seed data at start-up; trusted contacts, saved routes, community reports and journey records are written synchronously to a JSON file (.data/safecircle_store.json) after each change and reloaded on restart. The FastAPI service (Section 4.9) provides a PostgreSQL-backed data layer for the same entities. The store ships with two seeded community-report records (one marked "verified", one "pending") whose coordinates and content duplicate two of the demonstration incidents; because they are not flagged as demonstration data, they contribute to live scores through the report table and should be removed before deployment.

1.20. Emergency SOS and trusted contacts

SOS can be raised from the navigation bar or the dashboard. To prevent accidental activation, the SOS dialog requires the button to be held for two seconds, with haptic feedback, and then runs a three-second countdown during which the alert can be cancelled. When the countdown ends, the application sounds an alarm, can call the primary trusted contact automatically through a tel: URI, and posts the current coordinates to /api/sos/trigger. The server reverse-geocodes the position, resolves the district police control number, sends an alert containing the address and a map link to each trusted contact, and records a delivery status for each contact. The dialog also offers one-tap calls to each trusted contact, 112 and the district control room, pre-filled WhatsApp messages (wa.me links), cancellation of an active alert with a reason, and a simulated incoming "fake call" that a user can use to leave an uncomfortable situation. Trusted contacts (name, telephone number and relationship) can be added, edited and deleted.

1.21. Live trip monitoring and journey sharing

After choosing a route, the user can start live tracking, which uses watchPosition of the Geolocation API with high accuracy. On each position fix the client updates the map, estimates speed (from the reported speed or from successive fixes) and heading, computes the straight-line distance remaining to the destination, declares arrival within 50 m, and requests a new route analysis from the current position if the user is more than 150 m from the nearest route vertex (at most once every 15 s). A preview mode animates a marker along the route geometry at 500 ms per vertex to simulate navigation.

Trip status can be shared in two ways. The user can select a trusted contact as guardian and send WhatsApp messages at the start of the trip, as manual updates and on arrival, each containing a map link for the position at the time of sending. For continuous sharing, a journey session generates a time-limited, revocable link (/share/:token) through which a guardian can follow the traveller's current position, breadcrumb trail, progress, remaining distance and estimated arrival time until the journey ends or the link expires.

1.22. Battery, connectivity and offline support

On the dashboard and assistant views, a status component reads the Battery Status API where the browser supports it, the online/offline state and the connection type reported by the Network Information API. When the battery is at or below 20 % or the device is offline, it shows a power-saving prompt, keeps the last known position and context in localStorage, and offers an sms: link with a pre-composed message so that a location can be sent without mobile data; it also includes an offline alarm generated with the Web Audio API. For offline use, a service worker caches the application shell and visited map tiles in production builds, and an IndexedDB store keeps map tiles, recently analysed routes and safety data so that they remain available without a connection.

1.23. Administrator dashboard

Accounts with the administrator role have access to a dashboard with three areas: an overview of system health and summary analytics, a moderation queue in which community reports can be verified or rejected with notes, and an audit log of administrative actions.

1.24. Multilingual AI safety assistant

The assistant, available as a full page and as a floating voice widget, accepts typed or spoken questions in six languages. The server first resolves any place mentioned in the question (23 hard-coded alias patterns, the 110-entry gazetteer and district names; the Tamil-script forms in the alias patterns never match, because JavaScript's \b word boundary recognises only ASCII word characters), detects the language from Unicode script ranges, and assembles a context object containing the address, district, police control number, location score (Section 3.5), weather, traffic estimate and the nearest police station and hospital from the threat radar. Questions that match a multilingual keyword pattern (for example traffic, police, hospital, safety score, emergency, weather, lighting or battery) are answered from deterministic templates filled with this context, without calling an LLM. Other questions are sent to the Gemini API (model identifier gemini-3.1-flash-lite in the code, temperature 0.2, 4.5 s timeout) with a system instruction that contains the context and asks for an answer in the detected language. If no API key is configured, or if the call fails or times out, the template answer is returned. Of the 32 quick-prompt texts in the interface, 29 match the keyword pattern (the Telugu, Malayalam and Kannada "help" prompts do not), so most interactions in practice receive template answers. The route and weather/lighting templates exist only in Tamil and English; users of the other four languages receive English text for those intents.

Speech input runs the browser's Web Speech API and a MediaRecorder capture in parallel; if speech recognition yields no text, the recorded audio is sent to the server and transcribed by Gemini, trying a list of model identifiers in turn. Speech output truncates the text to 520 characters and first requests MP3 audio from an unofficial Google Translate text-to-speech endpoint (up to four chunks of about 170 characters), then falls back to Gemini

speech synthesis and finally to the browser's speechSynthesis. The safety scores are produced entirely by the rule-based engine; the LLM neither computes nor modifies them.

1.25. Reference Python backend

The repository also contains a FastAPI service with SQLAlchemy models for seven tables (users, trusted contacts, incidents, reports, saved routes, emergency events and audit logs) and Pydantic request schemas, which provides a PostgreSQL-backed data layer and interactive OpenAPI documentation. Coordinates are stored as floating-point columns; spatial types and indexes (PostGIS) are not yet used.

1.26. Implementation status

Table 4 summarises the implementation of each feature.

Table 4 - Feature implementation status.

Feature	Status	Notes
Tamil Nadu geofence and district resolution	Implemented	39-vertex boundary polygon; 16 district reference points
Geocoding and autocompletion	Implemented	Gazetteer, Photon and Nominatim; geofence-filtered
Alternative routes (OSRM)	Implemented	Three candidates per request, with fallbacks
Location risk score	Implemented	Rule-based model of Eqs. (1)–(7)
Route risk score and explanation	Implemented	Rule-based model of Eqs. (8)–(10); post-processed by Eq. (11)
Threat Radar map with provenance labels	Implemented	Severity circles; source labels; Demonstration Mode switch
Community hazard reporting	Implemented	Anonymous submission; administrator moderation
Weather context	Implemented	Open-Meteo current conditions
Traffic congestion	Implemented (heuristic)	Time-of-day estimate; no live traffic feed
SOS call and WhatsApp launch	Implemented	One-tap calls to contacts, 112 and district control; pre-filled WhatsApp messages
Automatic SOS alert delivery	Implemented	Server-side alert with location to each trusted contact; per-contact status
Trusted contacts	Implemented	Add, edit and delete
Saved routes	Implemented	Save and reload frequent routes
Live GPS tracking and rerouting	Implemented	Position tracking, off-route rerouting, arrival detection
Continuously updated location-sharing link	Implemented	Time-limited, revocable /share/:token link
Assistant (text)	Implemented	Templates for routine intents; Gemini for open questions (API key needed)
Voice input and output (six languages)	Implemented	Web Speech API with server transcription and speech fallbacks
Battery and connectivity guardian	Implemented	Battery, network and offline status; SMS fallback; offline alarm
Offline support	Implemented	Service worker and IndexedDB caching of tiles, routes and safety data
User authentication and roles	Implemented	Registration, sign-in; user and administrator roles
Administrator moderation dashboard	Implemented	Moderation queue, analytics, system health, audit log
Countdown SOS dialog and fake call	Implemented	Two-second hold, three-second cancellable countdown, alarm, simulated call
FastAPI / PostgreSQL backend	Implemented	SQLAlchemy models for seven tables; OpenAPI documentation

2. Testing and verification

This section reports the verification actually performed on the submitted source code: a build and static analysis, execution of the bundled unit tests, and deterministic re-execution of the scoring functions. The live external services (OSRM, Overpass, Open-Meteo, Nominatim and Gemini) were not reachable from the verification environment, so end-to-end tests with live data could not be run; the same condition exercised the fallback paths.

2.1. Build and static analysis

Dependencies install only with npm's --legacy-peer-deps option, because the declared esbuild version (^0.25) conflicts with the optional peer requirement of Vite 8 (esbuild ^0.27 or ^0.28). Once installed, the production build (vite build) completes successfully; Vite recommends splitting the single JavaScript bundle to reduce its size.

2.2. Unit tests

The repository contains one Python test file. Its eight test cases (four test functions, each also wrapped in a unittest class) all pass under pytest. However, they test a reference function defined inside the test file — with weights of 40 % incidents, 30 % lighting and 30 % emergency proximity and

a grey "insufficient data" class — rather than the TypeScript engine that the application uses, whose rules differ (Section 3). There are no automated tests of the running engine, the API or the user interface.

2.3. Deterministic check of the location model

Because the location model needs no external data, it was executed directly (Node.js 22, server time zone Asia/Kolkata) for six coordinates at 14:00 and 23:00 using the seeded data. Table 5 lists the results; each value follows from Eqs. (3)–(7). Values of $C(p)$ are rounded.

Table 5 - Output of the location model with the seeded data (obtained by executing the engine).

Location (latitude, longitude)	Time	C(p)	P(p)	τ	R	S	Level	Conf.
Gandhipuram bus stand, Coimbatore (11.0168, 76.9678)	14:00	12	0	1.00	12	88	LOW	88
	23:00	12	0	1.35	16	84	LOW	88
Cross Cut Road report site, Coimbatore (11.0195, 76.9654)	14:00	16	0	1.00	16	84	LOW	88
	23:00	16	0	1.35	22	78	MODERATE	88
R.S. Puram, Coimbatore (11.0081, 76.9516)	both	0	0	–	0	100	LOW	88
T. Nagar, Chennai (13.0418, 80.2341)	both	0	0	–	0	100	LOW	88
Thanjavur town (10.7870, 79.1378)	14:00	0	12	1.00	12	88	LOW	65
	23:00	0	12	1.35	16	84	LOW	65
Bengaluru, outside Tamil Nadu (12.9716, 77.5946)	both	–	–	–	–	–	OUT_OF_REGION	0

Two observations follow. First, locations without data receive the highest possible indicator (S = 100 at R.S. Puram and T. Nagar), and Thanjavur, whose nearest directory police station is about 49 km away, still scores 88 (LOW) with confidence 65: the model has no state that means "unknown". Second, because the time multiplier only scales existing risk, a location with no reports scores 100 at any hour.

2.4. Route model with external services unavailable

The route model was executed for a 41-vertex straight-line test geometry from Gandhipuram (11.0168, 76.9678) to R.S. Puram DB Road (11.0112, 76.9458) in Coimbatore — the endpoints of a seeded saved route — while all external services were unreachable. All 41 vertices were evaluated (k = 1). The collector fell back to the seeded facility directory (seven facilities in range) and the stored reports, and returned S_r = 93 at 23:00 and 96 at 14:00, both labelled "Lower Assessed Risk (Available Data)" with "sufficient" coverage and confidence 88. The seeded lighting report lay within 450 m of eight evaluated vertices; each encounter was counted both as a community-report encounter and as an "unlit road" encounter, although no OSM lit=no data had been retrieved. Had this been OSRM's first alternative, Eq. (11) would have displayed 96 in both cases. The test confirms that the fallback paths work, and shows that the coverage flag is driven by the seeded directory rather than by the availability of live data.

2.5. Aspects not yet evaluated

The following have not been carried out, and no results for them are claimed:

- user studies, usability tests or surveys — no user feedback is reported;
- comparison of the scores with any ground truth (recorded incidents, expert lighting audits or perceived-safety surveys) — no accuracy or correlation figures exist;
- end-to-end tests against the live routing, map-data, weather, geocoding and LLM services, latency measurements and load tests;
- documented tests of SOS calling and messaging on physical devices; and
- formal accessibility, security and privacy assessments.

Discussion

2.6. What the prototype demonstrates

The prototype shows that a safety-oriented route planner for an Indian state can be assembled entirely from open services and browser standards, without licence costs, and with server-side orchestration that keeps secrets off the client. The rule-based engine is small (about 330 lines of TypeScript), deterministic and fully specified by Eqs. (1)–(10), so every score can be traced to its inputs — a property that learned models do not offer without additional explanation methods. The provenance model, the exclusion of demonstration incidents from scoring and the decision not to turn district-level statistics into synthetic map points are sound practices that avoid common sources of misleading safety maps.

2.7. Validity of the safety indicator

The indicator measures exposure to a small set of mapped proxies; it does not measure crime risk. Its weights (for example 4.5 points per severity unit within 450 m, or a 6-point bonus for a facility within 800 m) were chosen by hand, and the lighting and CPTED literature (Section 2.2) supports the direction of the factors but not their magnitudes. Three properties of the current implementation further limit validity. First, missing data raise the score. OSM lighting and facility coverage is incomplete, so sparsely mapped areas receive the highest indicators, which contradicts the design principle that missing data must not be read as safety; the location model has no "unknown" state (Table 5). Second, the route model retrieves context once, around the route midpoint, so for long or inter-district routes most of the geometry lies outside the queried area; unlit roads are represented only by the centre points of their ways, and only four road classes are queried. Third, the rank-based banding of Eq. (11), together with the index-based route names and traffic summaries, overrides the computed scores and attaches claims — "well-lit", "lower patrol density", "high CCTV & patrol presence" — that are not derived from any data, while the coverage warning is overwritten and the confidence value is never shown. In this respect the build departs from its own "indicator, not guarantee" principle. Until this step is removed, the displayed ranking reflects OSRM's ordering rather than the safety model.

2.8. Accuracy of assistant outputs

In the assistant, when the context lookups exceed their time limits (1.2 s for reverse geocoding and 1.3 s for the threat radar), or when no facility is found, the code substitutes placeholder records — a police station named after the nearest gazetteer place or district headquarters and a hospital, reported at fixed distances of 620–890 m with the district control-room number or 108 — and presents them as the nearest facilities; a timed-out radar lookup also yields default weather ("Clear Visibility", 29 °C). These placeholders are cached for three minutes, and because the assistant queries the radar with a 7 km radius, it cannot reuse the 8 km entries already cached by route analysis, which makes timeouts on first queries more likely. The placeholder values and the heuristic traffic estimate are passed to the LLM as "verified live telemetry", and some templates state that a corridor has CCTV and patrol coverage. These are the conditions under which LLM answers become confidently wrong (Ji et al., 2023), and they must be corrected before any field use. A smaller issue of the same kind occurs in the interface, which shows a default score of 88 when a score is missing, and in the data-source panel, which can report a service as "online" when a cached request had failed.

2.9. Ethical, privacy, security and legal considerations

Safety maps can stigmatise neighbourhoods, and community reports can be mistaken or malicious (Goodchild, 2007). Because reports can be submitted anonymously, moderation is essential, and reports should influence scores and maps only after review. Report descriptions are inserted without HTML escaping into map pop-ups, so a malicious report could inject script that runs for every viewer of the Threat Radar (a stored cross-site scripting vulnerability). Personal data (contact names, telephone numbers and locations) must be protected, and the seed data include demonstration credentials stored in plain text that must be removed. Production use requires output escaping, rate limiting, data minimisation and compliance with India's Digital Personal Data Protection Act (2023). Public instances of the open services are provided under usage policies that limit request volumes (OpenStreetMap Foundation Operations Working Group, n.d.), and the primary text-to-speech path relies on an undocumented endpoint whose terms of use are unclear.

Limitations

The principal limitations of the current version are:

- heuristic, unvalidated weights and thresholds, with no ground truth against which to calibrate them;
- rank-based banding, index-based labels and route names that override the computed route scores (Eq. 11);
- missing data treated as favourable, and coarse, fixed confidence values that are not shown to users;
- context retrieved only around the route midpoint, unlit roads represented by way centre points, only four road classes queried for lighting, and Overpass output capped at 85 elements;
- time-of-day factors computed from the server clock, so a server running in Coordinated Universal Time would apply the night factor five and a half hours late;
- a heuristic traffic estimate rather than live traffic data;
- placeholder facilities and weather in assistant fallbacks, and default values shown when data are missing;
- no output escaping of user-submitted text and no rate limiting;
- a district registry covering 16 of 38 districts and a coarse 39-vertex boundary that admits parts of neighbouring territories;
- durations estimated from fixed speeds, and padding of the candidate set with synthetic arcs that do not follow roads; and
- unit tests that do not exercise the deployed engine.

Future scope

Corrections needed before any evaluation, in order of priority, are:

- remove the rank-based banding and the index-based labels, names and traffic summaries, rank routes by the computed S_r, and display the explanations, risk zones, coverage flag, confidence and per-segment colours that the API already returns;

- introduce an explicit "insufficient data" state driven by OSM tag coverage along the route, and stop substituting placeholder facilities, weather or default scores;
- escape all user-supplied text before rendering it and add rate limiting;
- label the traffic value as an estimate, or replace it with a licensed traffic feed;
- add automated tests for the scoring engine, the API and the SOS flow, including tests on physical devices;
- add PostGIS spatial types and indexes to the PostgreSQL backend;
- query context along the whole route corridor rather than around its midpoint, using way geometries rather than centre points and all relevant road classes; and
- compute time-of-day factors explicitly in Indian Standard Time and replace the geofence with an authoritative state boundary.

Research extensions include evaluating the indicator against ground truth — perceived-safety surveys of women travellers in Coimbatore and Chennai, field audits of street lighting and, where obtainable under appropriate agreements, official incident data; calibrating the weights on such data while keeping the model interpretable; conducting a usability study with target users under institutional ethics approval; and extending coverage to all 38 districts, public-transport stops and facility opening hours.

Conclusion

SafeCircle AI shows that open geospatial services, a compact rule-based engine and device-native communication channels can be combined into a route-planning and emergency-support platform for women travellers in Tamil Nadu, with SOS alerting, live journey sharing, community reporting with moderation, offline support and a multilingual assistant. Its strengths are transparent scoring rules, explicit data provenance and the exclusion of demonstration incidents from scoring. Its main weaknesses are also clear from the code: the displayed route ranking and labels are dominated by a rank-based adjustment, missing data are read as favourable, traffic is estimated heuristically, and the assistant can fall back to placeholder data. The scores have not been validated and should not be interpreted as measures of safety. With the corrections listed in Section 8, the system would provide a sound basis for the empirical evaluation with users and ground-truth data that the problem ultimately requires.

Acknowledgements

[Placeholder — acknowledge the project guide, department, institution and any funding.] Map data © OpenStreetMap contributors, available under the Open Database Licence (ODbL).

Appendix A. Scoring parameters of the evaluated build

Table A1 lists the numerical parameters of the location and route models as implemented in server/safetyEngine.ts and server/services/threatRadarCollector.ts, to support replication and future calibration.

Table A1 - Parameters of the location and route models.

Parameter	Value	Used in
Earth radius R_e	6 371 km	Eqs. (1)–(2)
Severity mapping	low 2, medium 3, high 4, critical 5 (default 3)	Eqs. (3)–(4), (8)
Incident radius, weight, recency window, floor	2 000 m; 6; 30 days; 0.2	Eq. (3)
Report radius, weight	1 500 m; 4	Eq. (4)
Isolation thresholds and penalties	2 000 / 4 000 m; 6 / 12; 10 if none	Eq. (5)
Night multipliers	1.35 (22–04 h); 1.15 (19–21 h, 05–06 h)	Eq. (6)
Location level thresholds	LOW ≥ 82; MODERATE ≥ 65; HIGH ≥ 45	Section 3.5
Location confidence	88 if facility within 10 km, else 65	Section 3.5
Base penalty b(h)	8 (19:00–05:59); 2 otherwise	Eq. (8)
Observation radius and weight	450 m; 4.5	Eq. (8)
Weather radius and weight	10 km; 3	Eq. (8)
Facility bonus; remoteness penalty	6 if ≤ 800 m; 5 if > 3 500 m	Eq. (8)
Vertex score bounds	[25, 98]	Eq. (9)
Sampling step	max(1, ⌊N/24⌋)	Section 3.6
Route level thresholds	LOW ≥ 80; MODERATE ≥ 55	Section 3.6
Per-vertex level thresholds	HIGH < 55; MODERATE < 78	Section 3.6
Routing timeouts	OSRM 6.5 s; via-points 4.5 s; fallback server none	Section 3.6
Synthetic padding distances	1.08× and 1.16× the primary route	Section 3.6
Risk-zone threshold	σ < 75 with ≥ 1 observation	Section 3.6
Route confidence	88 if any data, else 60	Section 3.6
Radar radius; cache lifetime	8 km; 10 min	Section 3.4
Radar inclusion radii: directory facilities; reports; demo incidents	2.5×; 3×; 3.5× the query radius	Section 3.4

Parameter	Value	Used in
Overpass radii	facilities 2–12 km; lit=no 5 km; construction 6 km	Section 3.4
Banding (evaluated build)	[88, 96], [78, 87], [68, 79] by OSRM index	Eq. (11)
Displayed level after banding	HIGH < 60; MODERATE < 82	Section 3.7
Default for a zero or missing route score	88 (before banding)	Section 3.7
Assistant lookup timeouts; context cache	1.2 s / 1.3 s; 3 min	Section 6.3

REFERENCES

- Borker, G. (2021). *Safety first: Perceived risk of street harassment and educational choices of women* (Policy Research Working Paper No. 9731). World Bank. https://doi.org/10.1596/1813-9450-9731
- Cáceres, M., & Grant, R. (Eds.). (2026). *Geolocation* [W3C Candidate Recommendation Snapshot, 26 March 2026]. World Wide Web Consortium. https://www.w3.org/TR/2026/CR-geolocation-20260326/
- Chalfin, A., Hansen, B., Lerner, J., & Parker, L. (2022). Reducing crime through environmental design: Evidence from a randomized experiment of street lighting in New York City. *Journal of Quantitative Criminology, 38*(1), 127–157. https://doi.org/10.1007/s10940-020-09490-6
- Cozens, P. M., Saville, G., & Hillier, D. (2005). Crime prevention through environmental design (CPTED): A review and modern bibliography. *Property Management, 23*(5), 328–356. https://doi.org/10.1108/02637470510631483
- *Digital Personal Data Protection Act, 2023*, No. 22 of 2023 (India). (2023). India Code. https://www.indiacode.nic.in/bitstream/123456789/22037/2/a2023-22.pdf
- Dijkstra, E. W. (1959). A note on two problems in connexion with graphs. *Numerische Mathematik, 1*, 269–271. https://doi.org/10.1007/BF01386390
- Eisenhut, K., Sauerborn, E., García-Moreno, C., & Wild, V. (2020). Mobile applications addressing violence against women: A systematic review. *BMJ Global Health, 5*(4), Article e001954. https://doi.org/10.1136/bmjgh-2019-001954
- Galbrun, E., Pelechrinis, K., & Terzi, E. (2016). Urban navigation beyond shortest route: The case of safe paths. *Information Systems, 57*, 160–171. https://doi.org/10.1016/j.is.2015.10.005
- Geisberger, R., Sanders, P., Schultes, D., & Delling, D. (2008). Contraction hierarchies: Faster and simpler hierarchical routing in road networks. In C. C. McGeoch (Ed.), *Experimental algorithms: 7th International Workshop, WEA 2008* (Lecture Notes in Computer Science, Vol. 5038, pp. 319–333). Springer. https://doi.org/10.1007/978-3-540-68552-4_24
- Gemini Team. (2023). *Gemini: A family of highly capable multimodal models* (arXiv:2312.11805). arXiv. https://arxiv.org/abs/2312.11805
- Goodchild, M. F. (2007). Citizens as sensors: The world of volunteered geography. *GeoJournal, 69*(4), 211–221. https://doi.org/10.1007/s10708-007-9111-y
- Haklay, M., & Weber, P. (2008). OpenStreetMap: User-generated street maps. *IEEE Pervasive Computing, 7*(4), 12–18. https://doi.org/10.1109/MPRV.2008.80
- Hormann, K., & Agathos, A. (2001). The point in polygon problem for arbitrary polygons. *Computational Geometry, 20*(3), 131–144. https://doi.org/10.1016/S0925-7721(01)00012-8
- Ji, Z., Lee, N., Frieske, R., Yu, T., Su, D., Xu, Y., Ishii, E., Bang, Y. J., Madotto, A., & Fung, P. (2023). Survey of hallucination in natural language generation. *ACM Computing Surveys, 55*(12), Article 248. https://doi.org/10.1145/3571730
- Kim, J., Cha, M., & Sandholm, T. (2014). SocRoutes: Safe routes based on tweet sentiments. In *Proceedings of the 23rd International Conference on World Wide Web (WWW '14 Companion)* (pp. 179–182). ACM. https://doi.org/10.1145/2567948.2577023
- Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. *Advances in Neural Information Processing Systems, 33*, 9459–9474. https://papers.nips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html
- Liu, E. (Ed.). (2026). *Web Speech API* [Draft Community Group Report]. W3C Audio Community Group. Retrieved October 29, 2026, from https://webaudio.github.io/web-speech-api/
- Loukaitou-Sideris, A. (2014). Fear and safety in transit environments from the women's perspective. *Security Journal, 27*(2), 242–256. https://doi.org/10.1057/sj.2014.9
- Luxen, D., & Vetter, C. (2011). Real-time routing with OpenStreetMap data. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 513–516). ACM. https://doi.org/10.1145/2093973.2094062
- Mata, F., Torres-Ruiz, M., Guzmán, G., Quintero, R., Zagal-Flores, R., Moreno-Ibarra, M., & Loza, E. (2016). A mobile information system based on crowd-sensed and official crime data for finding safe routes: A case study of Mexico City. *Mobile Information Systems, 2016*, Article 8068209. https://doi.org/10.1155/2016/8068209
- Maxwell, L., Sanders, A., Skues, J., & Wise, L. (2020). A content analysis of personal safety apps: Are they keeping us safe or making us more vulnerable? *Violence Against Women, 26*(2), 233–248. https://doi.org/10.1177/1077801219832124
- Ministry of Home Affairs. (n.d.). *Emergency Response Support System (ERSS)*. Government of India. Retrieved October 29, 2026, from https://www.mha.gov.in/emergency-response-support-system-erss
- National Crime Records Bureau. (2026). *Crime in India 2024*. Ministry of Home Affairs, Government of India. https://www.ncrb.gov.in/
- OpenStreetMap Foundation Operations Working Group. (n.d.). *Nominatim usage policy (aka geocoding policy)*. Retrieved October 29, 2026, from https://operations.osmfoundation.org/policies/nominatim/
- OpenStreetMap Wiki contributors. (n.d.). *Overpass API*. OpenStreetMap Wiki. Retrieved October 29, 2026, from https://wiki.openstreetmap.org/wiki/Overpass_API

- Quercia, D., Schifanella, R., & Aiello, L. M. (2014). The shortest path to happiness: Recommending beautiful, quiet, and happy routes in the city. In *Proceedings of the 25th ACM Conference on Hypertext and Social Media (HT '14)* (pp. 116–125). ACM. https://doi.org/10.1145/2631775.2631799
- Shah, S., Bao, F., Lu, C.-T., & Chen, I.-R. (2011). CROWDSAFE: Crowd sourcing of crime incidents and safe routing on mobile devices. In *Proceedings of the 19th ACM SIGSPATIAL International Conference on Advances in Geographic Information Systems* (pp. 521–524). ACM. https://doi.org/10.1145/2093973.2094064
- Shekar, A. (2019, December 22). 3 arrested for sexual harassment after Chennai woman alerts cops using Kavalan app. *The News Minute*. https://www.thenewsminute.com/tamil-nadu/3-arrested-sexual-harassment-after-chennai-woman-alerts-cops-using-kavalan-app-114589
- Sinnott, R. W. (1984). Virtues of the haversine. *Sky and Telescope, 68*(2), 158.
- Welsh, B. C., & Farrington, D. P. (2008). Effects of improved street lighting on crime. *Campbell Systematic Reviews, 4*(1), 1–51. https://doi.org/10.4073/csr.2008.13
- Zippenfenig, P. (2023). *Open-Meteo.com Weather API* [Computer software]. Zenodo. https://doi.org/10.5281/zenodo.7970649