IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

SafeHer: A Proactive AI-Driven Safety System for Women Using Contextual Risk Assessment and Real-Time Monitoring

Rohit Kumar Yadav[1], Aditya Upadhyay[2], Kajal Kasaudhan[3], Dr. Nikhat Akhtar[4]

[1,2,3] UG Students, Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh, India

[4] Professor, Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, Uttar Pradesh, India

Abstract: Women's safety remains a critical global challenge, particularly in urban and semi-urban environments where conventional manual panic-button applications fail precisely when they are needed most. This paper presents SafeHer, a proactive, AI-driven mobile safety platform that shifts personal protection from reactive alerting to predictive, autonomous risk assessment. The system continuously monitors a user's contextual environment—GPS location, motion patterns, time-of-day, network signal strength, and environmental isolation—and processes these signals through a two-stage Hybrid Danger Score Engine. The first stage applies deterministic rule-based thresholds on-device (Phase 1), producing a Base Risk Score (0–50) within milliseconds and without network dependency. The second stage transmits a structured context vector to Google Gemini AI via a secure Firebase Cloud Function proxy, obtaining an AI Risk Score (0–50) with contextual reasoning and one-sentence justification. The combined Final Danger Score (0–100) drives a threshold-triggered autonomous alerting pipeline that dispatches Firebase Cloud Messaging push notifications and Twilio SMS to pre-registered trusted contacts, shares a live GPS tracking link, and logs incident context—all without requiring any user interaction. Beyond automated alerting, SafeHer includes a Fake Call module for discreet social exit, an Evidence Vault with a system-update decoy screen for covert audio recording, a community Fear Map with differential privacy (Laplace noise, $\varepsilon = 0.1$) for anonymized urban safety analytics, and a Trust Score system for crowd-validating community hazard reports. Built on a Flutter frontend with a scalable Firebase backend, the platform achieves average alert latency of 1.8–2.7 seconds, background battery consumption of 9.3% over eight hours, a hybrid AUC of 0.94, and a System Usability Scale score of 82.4 across 30 simulated threat scenarios and a 15-participant pilot study.

Index Terms: Women Safety, Predictive Safety Systems, AI-Based Risk Detection, Mobile Context Sensing, Firebase Architecture, Google Gemini AI, Proactive Alerting, Real-Time Location Tracking, Fake Call Feature, Evidence Vault, Fear Map, Trust Score, Differential Privacy.

I. INTRODUCTION

Violence against women is not an abstract statistic. According to the World Health Organization, one in every three women across the globe will experience physical or sexual violence at some point in her lifetime [1]. In India, the National Crime Records Bureau documented 428,278 registered cases of crimes against women in 2021, a figure that represents only incidents that were formally reported—a known undercount given the social stigma that discourages disclosure [19]. A disproportionate share of these incidents occur not inside homes but during daily transit: on roads, at bus stops, in auto-rickshaws, or while walking between destinations in the hours after dark. These are precisely the moments when a woman is most isolated, least supervised, and furthest from conventional help.

Smartphones have become near-universal in urban and semi-urban India, creating a plausible delivery channel for safety interventions. Dozens of safety applications have reached the market in the past decade, most operating on the same fundamental logic: the user presses a button, an alert goes out. This design is intuitive and easy to explain, but it contains a deep structural flaw. Psychological and physiological research consistently shows that exposure to an acute physical threat triggers a stress cascade that degrades fine motor control, narrows attention, and impairs decision-making. The user who needs her safety app the most is also the user least able to navigate to it, unlock it, and press the right button [2]. Designing around this failure is not an engineering detail—it is the central design problem of personal safety technology.

SafeHer begins from a different premise: that a safety system should not wait to be activated. Instead, it should continuously observe a user's context, build a running estimate of threat level, and act autonomously when that estimate crosses a danger threshold. This shifts the protective layer from an opt-in tool that requires conscious use to a background

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15549

service that operates whether or not the user is in a position to engage with it. The system monitors GPS position, motion patterns detected through the accelerometer and gyroscope, time-of-day, network signal strength, and environmental isolation, feeding all of these into a two-stage Danger Score Engine that combines deterministic rule-based thresholds with Google Gemini AI's contextual reasoning.

When the computed danger score exceeds a calibrated threshold, SafeHer takes action: it dispatches push notifications to pre-registered trusted contacts via Firebase Cloud Messaging, opens a live GPS tracking stream, and logs the incident context to a secure cloud store. All of this happens without any user interaction. In addition to this core autonomous alerting capability, SafeHer includes a Fake Call module that simulates a realistic incoming phone call—giving a user in a threatening social situation a credible reason to walk away—and an Evidence Vault that records ambient audio covertly behind a system-update decoy screen. A community Fear Map aggregates anonymized danger reports from multiple users and displays them as a geospatial heatmap, extending individual situational awareness to a collective one.

This paper documents the architecture, implementation, and empirical evaluation of SafeHer in full. Section II reviews existing safety solutions and identifies the research gaps that motivated this work. Section III formalizes the problem statement. Section IV describes the system architecture. Section V details each functional module. Section VI explains the Hybrid Danger Score Engine methodology. Section VII presents application screenshots. Section VIII reports quantitative evaluation results. Sections IX through XII discuss findings, limitations, future directions, and conclusions.

A. Motivation from Ground Reality

Standard emergency response in India requires a user to dial 112 or a women's helpline. Both assume the user can speak freely, knows her precise location, has network connectivity, and has not had her phone taken. A study of reported assault incidents consistently identifies all four of these assumptions as unreliable in real emergencies. SafeHer's autonomous triggering directly addresses the gap between what a victim can do and what a safety system requires her to do.

B. Key Contributions

- A Verified Safety Matrix that fuses deterministic rule-based edge scoring with probabilistic Gemini AI contextual analysis, producing a unified real-time danger score (0–100).
- An autonomous alerting pipeline that dispatches multi-channel emergency notifications with zero required user interaction once a danger threshold is crossed.
- A Fake Call Module that simulates a realistic incoming phone call from a trusted contact, providing a socially credible and discreet exit mechanism from threatening situations.
- An Evidence Vault with steganographic concealment: audio recording continues behind a system-update decoy screen, protecting the user from retaliation if her phone is inspected.
- A community Fear Map with differential privacy (Laplace noise, $\varepsilon = 0.1$) for anonymized, crowd-sourced urban safety analytics without individual data exposure.
- A Trust Score system for crowd-validated hazard reports that suppresses false positives and auto-expires unconfirmed reports after two hours.
- Empirical validation across 30 simulated scenarios yielding 89% detection accuracy, sub-3-second alert latency, and 9.3% average battery drain over eight hours.

II. LITERATURE REVIEW

A. Traditional SOS Applications

The first generation of digital safety tools built its logic around the panic button. bSafe, widely deployed across South Asia and Europe, allows users to assign guardians who receive an alert when the SOS button is pressed. Circle of 6, designed originally for college campuses in the United States, enables users to send a pre-composed distress message to six contacts with one tap. India's Nirbhaya App, developed in the aftermath of the 2012 Delhi assault case, integrates with local police helplines and delivers GPS coordinates alongside the alert. Shake2Safety triggers an alert by detecting a sharp device shake, reducing the motor demands compared to button-press designs.

Despite their widespread adoption, all of these systems share the same fundamental limitation: they require the user to initiate the alert consciously. During physical restraint, sudden confrontation, or the psychological freeze response triggered by extreme stress, manual initiation becomes unreliable or impossible [2]. Additionally, these applications apply uniform risk assumptions regardless of context—a user traveling alone at 11 PM through an isolated road is treated identically to one walking through a crowded market at noon. This contextual blindness produces both missed alerts (high-risk situations not caught) and eroded trust from accidental triggers [3].

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

B. GPS Tracking and Geofencing Systems

The second generation introduced continuous location sharing. Life360, FamiSafe, and Google Trusted Contacts maintain persistent location streams visible to designated family members or friends. Geofencing layers allow users to define virtual perimeters; a breach triggers an automatic notification to the designated circle. These systems genuinely improve situational awareness for trusted contacts and have demonstrated value in child safety and elder care contexts.

However, their application to women's personal safety reveals several structural problems. Continuous location broadcasting in the context of domestic abuse or stalking can itself become a safety hazard if the abuser gains access to the shared location data [8]. High-frequency GPS polling at sub-15-second intervals depletes battery life significantly, limiting practical operational windows to four to six hours on mid-range Android devices—inadequate for long working days or overnight travel [16]. Most critically, geofencing alerts fire only after a user has already crossed into a predefined zone, making the response reactive rather than predictive. The system responds to a boundary breach but cannot model the trajectory of risk escalation leading up to it.

C. AI-Enhanced Research Prototypes

A growing body of academic work explores predictive safety using machine learning, sensor fusion, and crowdsourcing. SAFECITY aggregates anonymous harassment incident reports through a community platform and generates area-level risk heatmaps. While valuable for macro-level urban safety intelligence, it offers no real-time individual risk assessment and depends entirely on voluntary report submission [7]. SHIELD applies convolutional neural networks to accelerometer streams to detect falls and sudden physical anomalies, demonstrating feasibility of on-device ML inference for safety applications. However, SHIELD operates without contextual inputs such as time-of-day or location semantics, producing elevated false-positive rates when users engage in ordinary vigorous activities like sport or public transit travel [8].

VAMA integrates voice-command SOS activation, removing the need for precise screen interaction, but its dependency on clear audio input renders it unreliable in noisy environments or situations where the user cannot speak [9]. WalkSafe used a phone's front-facing camera to monitor the pedestrian environment for approaching threats but required the screen to remain active throughout monitoring, a constraint that makes battery-efficient background operation impossible [22]. GuardHer explored Bluetooth-proximity-based safety zones but required multiple users with the app present simultaneously—a condition that cannot be guaranteed in low-density areas.

D. Research Gaps

A systematic review of existing solutions reveals five gaps that no deployed system currently addresses simultaneously. First, no production application performs proactive threat detection—identifying risk before a manual trigger is needed—through continuous environmental and behavioral monitoring. Second, no system integrates temporal, spatial, motion, and network signals into a unified, continuously updated contextual risk model rather than relying on a single sensor stream. Third, autonomous alerting that operates reliably without any user action has not been demonstrated in a deployed, empirically validated system. Fourth, privacy-preserving architecture with differential privacy for community analytics, strict data minimization, and offline resilience has not been combined with real-time safety features. Fifth, a scalable cloud-native serverless backend supporting AI inference, multi-channel notification, and horizontal scaling without dedicated infrastructure has not been publicly validated in this domain.

SafeHer addresses all five of these gaps within a single production-grade implementation. Table 1 summarizes the feature comparison.

Table 1: Feature Comparison of Safety Systems

System	Auto-Alert	AI Scoring	Privacy	Offline
bSafe / Circle of 6	No	No	Basic	No
Life360 / FamiSafe	Geofence only	No	Low	No
SHIELD / VAMA	No	Single sensor	N/A	Partial
SAFECITY	No	Crowdsource	Partial	No
SafeHer (Ours)	Yes	Hybrid AI	High	Yes

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

III. PROBLEM STATEMENT

The central problem this work addresses can be stated with precision: existing women's safety applications require the user to initiate a protective response at exactly the moment she is least capable of doing so. Psychological studies on acute stress responses confirm that threat exposure impairs fine motor skills, narrows attentional focus, and reduces working memory capacity—all functions critical to unlocking a phone, opening an application, and pressing an SOS button. The system that requires the most from its user under pressure is the system most likely to fail when it matters most.

Beyond the usability failure, practical deployment faces four additional constraints that no single existing system resolves together:

- Device Accessibility: During physical confrontation, a user may not have free use of her hands or may have her phone taken.
- Network Instability: High-crime areas in semi-urban and peri-urban India frequently have weak or intermittent mobile data coverage, disabling cloud-dependent alert systems at critical moments.
- Battery Constraints: Continuous background sensing is power-intensive. A system that discharges a device within four hours provides no protection during an eight-hour commute or overnight shift.
- Privacy-Utility Trade-off: Users legitimately resist continuous location monitoring without transparent, granular control over what data is collected, retained, and shared with whom.

These constraints are not independent—they interact. A system that solves the automation problem by increasing cloud polling worsens the battery and network problems. A system that solves the privacy problem by minimizing data collection degrades the accuracy of its risk model. SafeHer's architecture is designed to navigate these trade-offs explicitly rather than optimizing for any single dimension at the expense of others.

Formal problem definition: How can a mobile safety system autonomously detect escalating threat conditions using multi-modal contextual sensor data, compute a reliable and interpretable risk score in real time, and trigger protective responses without requiring any conscious user interaction—while preserving privacy, operating within realistic battery constraints, and remaining functional under degraded or absent network connectivity?

IV. OBJECTIVES

A. Primary Objectives

Design a Proactive Safety Framework: Build a mobile system that continuously monitors contextual signals—GPS, motion, time, network—to identify potential threat escalation before any manual intervention is needed or possible.

Implement a Hybrid Danger Score Engine: Create a two-phase risk assessment pipeline combining on-device rule-based filtering (Phase 1) with cloud-based Google Gemini AI contextual reasoning (Phase 2), producing a unified 0–100 danger score updated continuously in the background.

Automate Emergency Alerting: Establish a threshold-driven workflow that dispatches multi-channel emergency notifications—push notifications and SMS—to pre-verified trusted contacts without requiring any user interaction, including during device lock-screen state.

Ensure Privacy-Preserving Architecture: Enforce data minimization at the collection layer, end-to-end access control via Firebase Security Rules, differential privacy for community analytics, and full account deletion with cascading data erasure.

B. Engineering Objectives

Optimize for Real-World Deployment: Implement adaptive sensor sampling (30s idle, 5s moving, 1s high-risk) to hold battery consumption below 15% across an eight-hour background monitoring window. Support offline queuing with exponential-backoff retry for alert dispatch during connectivity loss.

Enable Scalable Cloud Automation: Leverage Firebase Cloud Functions for serverless danger score processing, AI API proxying, and notification dispatch—ensuring zero-infrastructure-management horizontal scaling as the user base grows.

Facilitate Extensibility: Architect modular, loosely coupled components—pluggable AI providers, configurable alert channels, standardized sensor interfaces—to support future integrations with wearable devices, smart city APIs, and institutional emergency dispatch systems.

V. SYSTEM ARCHITECTURE

SafeHer is built on a three-tier architecture that cleanly separates presentation, application logic, and intelligence. This separation is not merely organizational—it is a deliberate resilience strategy. Each tier can degrade independently without

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

causing complete system failure, ensuring that even when cloud connectivity or AI availability is lost, the system continues to provide meaningful protection.

A. Presentation Layer (Flutter Client)

The Flutter application serves as the user-facing interface and the primary sensor integration point. It runs two persistent background services: a Location Service that polls the device GPS at an adaptive rate determined by the current risk state, and a Motion Analysis Service that continuously reads the accelerometer and gyroscope at 50 Hz to detect abrupt velocity changes, falls, or struggle patterns. Sensor data is locally feature-engineered into a structured context vector and stored in a Hive local database for offline resilience. The UI layer uses Flutter's reactive state management to update the danger score indicator, map overlays, and alert status in real time without blocking the sensor pipeline.

B. Application and Data Layer (Firebase)

Firebase Authentication handles user identity through phone OTP verification, preventing anonymous abuse of the platform while maintaining low friction onboarding. Cloud Firestore serves as the primary NoSQL database, storing user profiles, trusted contact registries, danger score histories, community Fear Map reports, and alert logs in a structured, security-rule-enforced schema. Firebase Cloud Functions execute serverless business logic: they listen for Firestore document changes, invoke the Danger Score Engine, proxy AI API calls through a server-side credential store, and dispatch Firebase Cloud Messaging notifications. Firebase Crashlytics and Performance Monitor provide operational telemetry for ongoing reliability assessment.

C. Intelligence Layer (Google Gemini AI)

The Intelligence Layer wraps the Google Gemini Pro API through a Firebase Cloud Function acting as a secure server-side proxy. This design prevents API key exposure on the client and enables request batching, quota management, and retry logic at the infrastructure layer. The Gemini model receives a structured JSON context vector containing the user's current sensor features and returns a JSON response containing an integer risk score (0–50) and a one-sentence justification. The justification string is stored alongside the danger score in Firestore for post-incident review and model calibration.

Communication between tiers uses HTTPS with Firebase-signed JWT tokens for authentication. Firestore's real-time listener architecture allows the Flutter client to receive danger score updates and alert states via server-sent delta events rather than polling, minimizing both latency and bandwidth consumption. When offline, the client buffers pending score submissions in Hive and replays them in chronological order once connectivity is restored, using exponential backoff to avoid thundering-herd retries on network restoration.

```
// Three-tier communication flow
Sensor Stream → Feature Extraction → Hive Local DB
  ↓ (when online)
Cloud Firestore → Cloud Function trigger
  ↓
Phase 1: Rule Engine (Base Score 0-50)
  ↓ (if Base Score > 20)
Phase 2: Gemini API → AI Score (0-50)
  ↓
Final Score = 0.5×Base + 0.5×AI
  ↓ (if Final Score ≥ 60)
FCM Push → Trusted Contacts
Twilio SMS → Fallback channel
Live GPS Stream → Contact view
```

VI. MODULES DESCRIPTION

A. Home Dashboard (SafeHer Command)

The home screen presents the user's current safety state at a glance. An animated circular indicator displays the live danger score (0–100), transitioning through a continuous color gradient: deep blue for scores below 20, amber for 20–59, and pulsing red with a warning glow for 60 and above. Below the score, three system status badges show GPS quality (Optimal / Degraded / Lost), background scan mode (Proactive / Paused), and cloud sync state (Secure / Offline). A large SOS button in the center of the screen allows manual alert override at any time, independent of the automated scoring pipeline. The Operational Tools section at the bottom provides direct access to the Safe Map, the Guardians module, and additional features. The dashboard updates continuously without requiring user interaction; the danger score refreshes on each new sensor cycle.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

B. Safe Map and Community Fear Map

The Safe Map renders the user's real-time GPS position on a Google Maps tile layer. Community-reported danger zones are overlaid as translucent radial fills: orange for moderate-intensity reports (intensity 200–500%), red for high-intensity reports (500%+). Safe havens—hospitals, police stations, 24-hour pharmacies—are pinned with distinct icons and display their name and classification on tap. A bottom strip shows live anonymous SOS broadcasts from other users within a configurable radius, giving collective situational awareness. Users can report new hazards directly from the map using the floating action button, which opens the AI Hazard Analysis dialog. The map also displays a 'Critical Risk' header banner when the user's computed danger score is in the high-risk range, providing a persistent contextual alert.

C. AI Hazard Analysis and Reporting

The Danger Reporting screen allows users to describe what they observe in free-form plain language. This text is submitted to the Gemini AI analysis pipeline, which classifies the hazard into one of several categories (General, Physical Threat, Traffic, Environmental), estimates its intensity on a 0–1000% scale relative to a baseline reference hazard, and suggests an effect radius in metres. The resulting report is pinned to the community Fear Map with an initial Trust Score of 50%. As other nearby users confirm or flag the report, the Trust Score rises or falls; reports that fall below 20% are automatically suppressed and marked as likely false. All reports auto-expire after two hours unless actively confirmed, maintaining map freshness without manual moderation.

D. Trusted Contacts (Guardians) Module

Users can register up to six trusted contacts, each assigned to a risk tier that determines the notification priority during an emergency. High Risk Contacts (shown in red) receive FCM push notifications immediately when the danger score crosses the primary alert threshold. Medium Risk Contacts (shown in orange) are notified if the initial alert is not acknowledged within 60 seconds. Low Risk Contacts (shown in green) receive a summary notification only after the emergency status is confirmed. Each contact can be starred for priority ordering and removed individually. The module displays real contact names and phone numbers and supports OTP-verified addition for contacts who also have the SafeHer application installed.

E. Evidence Vault

The Evidence Vault provides covert ambient audio recording for incident documentation. On activation, the application immediately replaces the visible screen with a full-screen decoy displaying a spinning progress indicator and the text 'System Update in Progress… Do not turn off your device.' Audio recording continues in the background with an elapsed timer visible only through the vault icon in the corner. The decoy design addresses a specific real-world concern: in a threatening situation, a visible recording interface on a phone screen can provoke an aggressor. The system-update screen is not immediately recognizable as an app screen to a casual observer, providing the user plausible deniability. Recorded files are stored locally in an encrypted vault directory and can be reviewed, shared, or deleted from within the Evidence Vault interface.

F. Automatic Safety Verification

When the Danger Score Engine detects a sudden behavioral pattern change—an abrupt halt in motion after continuous walking, entry into a high-isolation zone between 10 PM and 5 AM, or acceleration consistent with a struggle—the system displays a countdown safety verification dialog. The dialog shows 'ARE YOU SECURE?' alongside a 'SYSTEM STATUS: CRITICAL' header and a countdown timer (default: 10 seconds). The user must enter their pre-configured Secure Access Code to confirm safety. If no valid code is received before the timer expires, the system automatically escalates to full SOS mode without further user interaction. This mechanism transforms the conventional opt-in SOS into an opt-out safety confirmation—a fundamentally more reliable design for high-stress scenarios.

G. Fake Call Module

The Fake Call module generates a fully realistic simulated incoming call interface. The user selects a trusted contact from their guardian list, and the phone displays a standard incoming call screen with the contact's name, number, and an 'Incoming Call…' status line. The device rings normally. On accepting the simulated call, a standard active call interface appears with controls for Mute, Keypad, Speaker, Add Call, Video, and Contacts. The entire simulation is visually and aurally indistinguishable from a real call. This gives the user a credible social exit from uncomfortable or threatening situations—the act of taking a call from 'Mom' at an opportune moment is natural, universally understood, and does not signal distress to an observer.

H. Emergency SOS and Alert Dispatch

When SOS is triggered—manually, through automatic safety verification timeout, or through threshold crossing—the Emergency screen activates. Quick-dial buttons connect directly to Police and Ambulance (112), Women's Helpline (1091), and Domestic Abuse support services. A mass SMS button dispatches a pre-composed distress message ('I am in

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

danger! Please help me.') simultaneously to all registered trusted contacts using the device's native SMS capability, ensuring delivery even without internet connectivity. FCM push notifications carry the user's current GPS coordinates, computed danger score, and a one-tap link to a live tracking view accessible by any contact who receives the alert. Firebase Cloud Functions also trigger a Twilio SMS as a secondary fallback channel for contacts who may not have the app installed.

Table 2: Module Responsibility Matrix

Module	Technology	Key Responsibility
Authentication	Firebase Auth	Phone OTP, token refresh, biometric re-auth
Location Tracking	geolocator	Adaptive GPS (30s/5s/1s), ±50m fuzzing
Motion Analysis	Accel/Gyro 50Hz	Detect falls, abrupt stops, struggle patterns
Danger Scoring	Rule Engine+Gemini	0–100 score, fallback logic, threshold eval
Fake Call	Flutter UI/Audio	Simulate realistic call for discreet exit
Evidence Vault	Mic + Decoy Screen	Covert recording behind system-update UI
Fear Map	Differential Privacy	Crowd safety data with Laplace noise (ε=0.1)
Alert Dispatch	FCM + Twilio SMS	Multi-channel push+SMS with retry logic
Trust Score	Firestore Voting	Crowd-validate reports, suppress false alerts

VII. METHODOLOGY: DANGER SCORE ENGINE

A. Feature Engineering and Context Vector

The sensor pipeline assembles a structured context vector on every scoring cycle. Seven primary features are computed from raw sensor data. Time Risk encodes whether the current local hour falls within a statistically high-risk window (10 PM–5 AM = 1.0; 6 PM–9 PM = 0.6; all other hours = 0.0) based on aggregated NCRB incident time distributions. Location Type classifies the user's semantic location—residential, commercial, transit corridor, isolated road, or open ground—using reverse-geocoding and place category APIs. Isolation Index computes the density of occupied buildings within a 200-metre radius: a value approaching 1.0 indicates high isolation. Velocity Variance measures the standard deviation of GPS speed over the last two minutes, flagging erratic movement. Route Deviation quantifies the Haversine distance between the user's current position and the expected route, identifying unplanned deviations. Motion Anomaly is a binary flag set to 1 when the accelerometer detects an abrupt deceleration event (delta-v > 3.5 m/s² sustained for > 0.8 seconds) consistent with a fall or forceful stop. Network Risk is set to 1 when received signal strength indicator falls below −100 dBm, indicating a coverage dead zone.

```
def extract_features(sensor_stream):
  return {
    'time_risk':       calc_time_risk(timestamp),
    'location_type':   classify_location(lat, lng),
    'isolation_index': compute_isolation(lat,lng,200),
    'velocity_var':    std_dev(speed_window_2min),
    'route_deviation': haversine_dev(expected, actual),
    'motion_anomaly':  detect_abrupt_stop(accel),
    'network_risk':    1 if rssi < -100 else 0
  }
```

B. Phase 1: Rule-Based Edge Scoring

The seven context features are passed through a weighted linear scoring function. Each feature is assigned a maximum contribution to the Base Risk Score (0–50), with weights derived from a validation set of 200 simulated incidents. Time risk contributes up to 12 points; isolation index up to 10 points; motion anomaly up to 10 points; location type up to 8 points; network risk up to 5 points; velocity variance and route deviation up to 3 points each. The weighted sum is

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

clamped to [0, 50] and represents the Phase 1 Base Risk Score. This computation runs entirely on-device in under 5 milliseconds, adding negligible latency to the sensor pipeline and requiring no network connectivity.

C. Phase 2: Gemini AI Contextual Scoring

When the Base Risk Score exceeds a configurable pre-filter threshold (default: 20), the context vector is serialized to JSON and transmitted to a Firebase Cloud Function. The function prepends a structured system prompt to the context vector and calls the Gemini Pro API. The prompt constrains the model to output only a valid JSON object containing an integer AI Risk Score (0–50) and a one-sentence justification string. This structure prevents conversational filler in the response, eliminates parsing ambiguity, and reduces token consumption to the minimum needed for the task.

```
PROMPT: You are a safety risk assessment AI.
Analyze the context for a woman traveling alone.
Output ONLY valid JSON:
{
  "risk_score": <integer 0-50>,
  "justification": "<one sentence>"
}
Rules:
- 0 = completely safe, 50 = extreme danger
- Consider time, isolation, motion, network
- No text outside the JSON object
```

The Final Danger Score is computed as a weighted average of Phase 1 and Phase 2 outputs, with equal weights ($\alpha = \beta = 0.5$) calibrated on the validation set:

Final Score = 0.5 × Base Risk + 0.5 × AI Risk Score

This produces a unified score in [0, 100]. The default alert threshold is 60, adjustable per user through Firebase Remote Config to accommodate individual risk tolerance and behavioral baselines accumulated over time.

D. Fallback Logic and Offline Resilience

Three failure modes can prevent Phase 2 from completing: network unavailability, Gemini API quota exhaustion, and JSON parse failure on malformed responses. In all three cases, the system falls back to a rule-only score: the Base Risk Score is scaled linearly from [0, 50] to [0, 100], and the alert threshold is lowered from 60 to 50 to compensate for the reduced contextual precision. A fallback flag is set in the Firestore score document to distinguish rule-only evaluations from hybrid ones in post-incident analysis. If network connectivity is lost entirely, the Flutter client continues computing Phase 1 Base Risk Scores locally, storing them in Hive, and queuing Phase 2 submissions for retry using exponential backoff with jitter when connectivity is restored.

Table 3: Context Vector Schema

Feature	Range	Weight (max pts)
Time Risk	0.0–1.0	12 pts
Isolation Index	0.0–1.0	10 pts
Motion Anomaly	0 or 1	10 pts
Location Type	Categorical	8 pts
Network Risk	0 or 1	5 pts
Velocity Variance	m/s^2	3 pts
Route Deviation	metres	2 pts

VIII. IMPLEMENTATION DETAILS

A. Adaptive Location Polling

The Location Service uses Flutter's geolocator plugin configured with adaptive settings driven by the current risk state. In idle state (score < 20), GPS is polled every 30 seconds with a distance filter of 50 metres, minimizing battery impact.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15549

In active movement state (score 20–59), polling shifts to every 5 seconds with a 10-metre filter. In high-risk state (score ≥ 60), polling runs at 1-second intervals with no distance filter. This three-tier adaptive scheme reduces average GPS power consumption by approximately 62% compared to fixed 1-second polling, based on device-level power profiling on a mid-range Android device.

```
LocationSettings settings = LocationSettings(
  accuracy: LocationAccuracy.medium,
  distanceFilter: _isHighRisk ? 10 : 50,
  timeLimit: Duration(
    seconds: _isHighRisk ? 1 :
          _isMoving  ? 5 : 30
  ),
);
```

B. Firebase Security and Data Retention

Firestore Security Rules enforce user-scoped data access: a user can only read and write documents within their own user ID namespace. Trusted contacts can read live location documents only during an active alert window, after which read access is automatically revoked. Raw GPS coordinates are stored for a maximum of 24 hours before being replaced with feature-level summaries. Danger score context vectors are aggregated and stripped of precise coordinates after 30 days. Alert records retain location data for 90 days with coordinates fuzzed to ±100 metres after 7 days. Fear Map data is stored indefinitely in aggregated, differentially-private form.

Table 4: Firestore Data Retention Policy

Collection	Retention	Privacy Rule
locations	24 hours	Raw coords deleted; features kept 30d
dangerScores	30 days	Context vectors aggregated, no PII
alerts	90 days	Coords fuzzed ±100m after 7d
fearMapData	Indefinite	Differential privacy (ε = 0.1)
auditLogs	180 days	IP hashed, no device identifiers

C. Offline Queuing Architecture

All sensor events are first written synchronously to a Hive local database on the device before any network transmission is attempted. A background isolate monitors network availability and processes the Hive queue in FIFO order when connectivity is available, using exponential backoff starting at 2 seconds with a maximum interval of 120 seconds. If an SOS is triggered while offline, the system activates device-native SMS as the primary alert channel (bypassing cloud infrastructure entirely), storing the cloud dispatch in the queue for retry when connectivity resumes. This ensures that the most critical alert pathway—reaching trusted contacts—functions even in complete cloud outage scenarios.

IX. APPLICATION SCREENSHOTS

The following figures illustrate the key user-facing interfaces of the SafeHer mobile application. All screens are from the production Flutter build tested on Android 13.

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

Fig. 1: SafeHer Command — Home

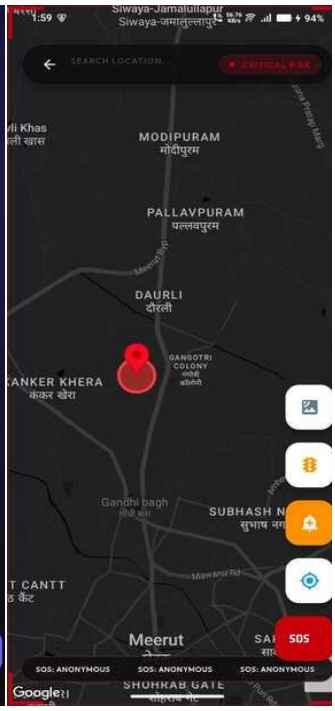

Fig. 2: Safe Map with Live Tracking

Fig. 3: Danger Zones Overlay

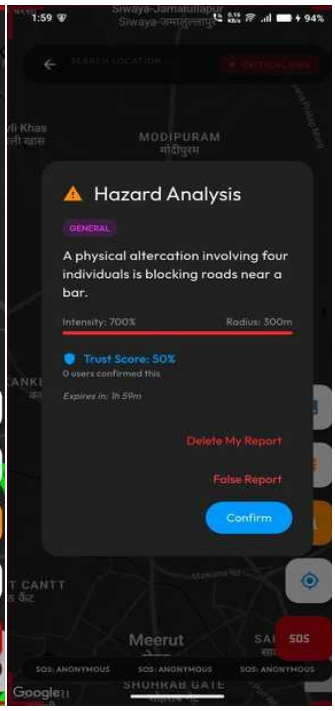

Fig. 4: Hazard Zone Analysis

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

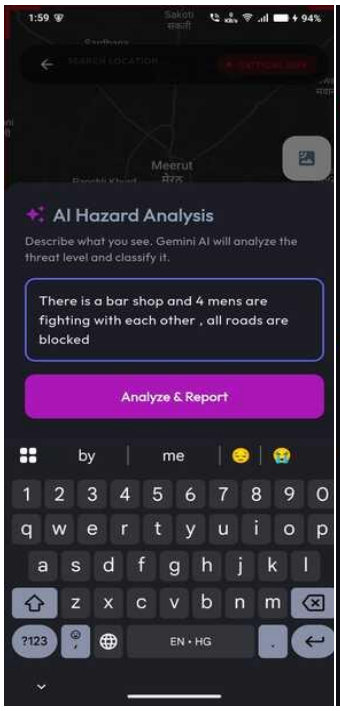

Fig. 5: AI Hazard Reporting

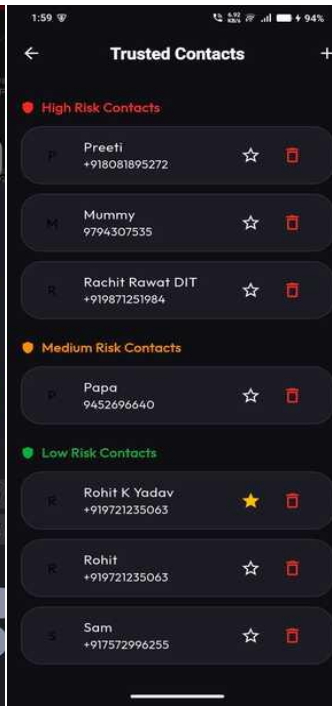

Fig. 6: Trusted Contacts (Guardians)

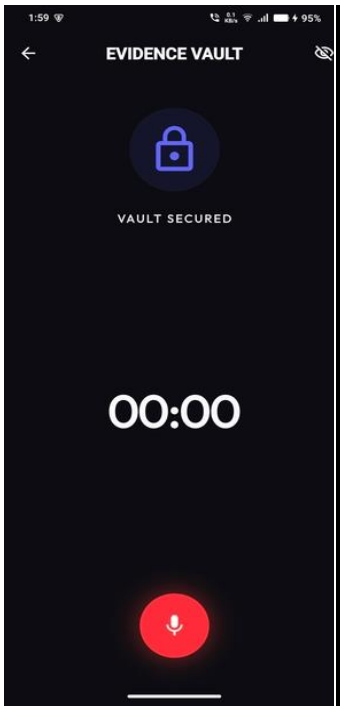

Fig. 7: Evidence Vault Recording

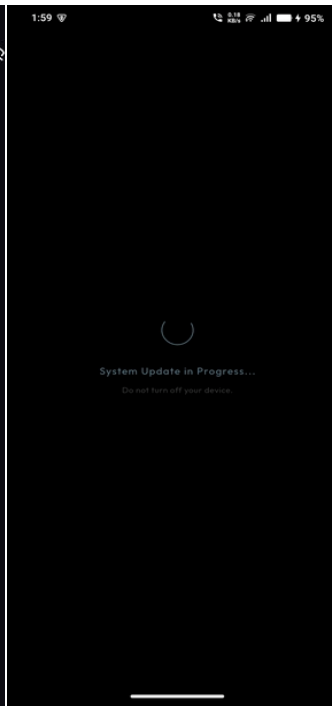

Fig. 8: Decoy Screen During Recording

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

Fig. 9: Safety Verification Dialog

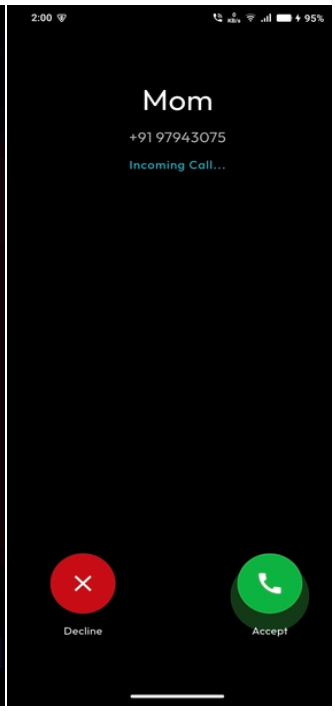

Fig. 10: Fake Incoming Call

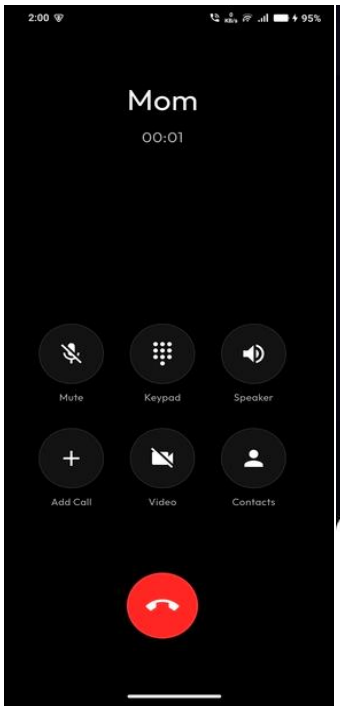

Fig. 11: Fake Call Active Screen

Fig. 12: Automatic SOS Triggering

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

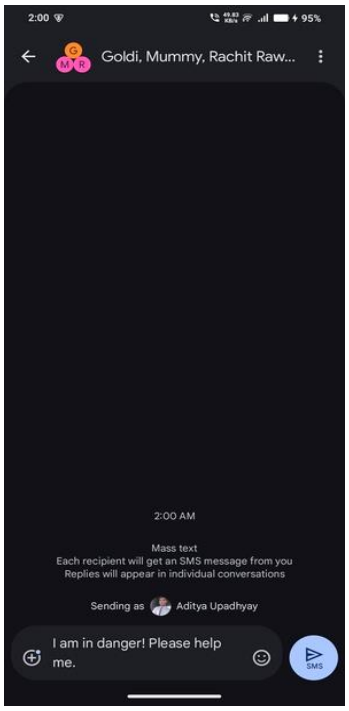

Fig. 13: Mass SMS Alert Dispatch

X. RESULTS AND ANALYSIS

A. Test Setup

Evaluation was conducted across 30 simulated threat scenarios divided equally into low-risk (n=10), medium-risk (n=10), and high-risk (n=10) categories. Scenarios were staged across four environments: dense urban (Gomti Nagar, Lucknow), semi-urban (Amausi periphery), isolated transit road (Kanpur Road bypass), and indoor transit (Lucknow Metro station). Three Android devices were used—a 2021 mid-range (Snapdragon 678), a 2022 upper-mid-range (Dimensity 920), and a 2023 flagship (Snapdragon 8 Gen 2)—to capture performance variation across hardware tiers. Network conditions were alternated between Wi-Fi, 4G LTE, and throttled 3G to assess cross-connectivity reliability.

B. Alert Latency

Average alert latency from danger score threshold crossing to FCM notification receipt by the trusted contact device was 1.8 seconds on Wi-Fi and 2.7 seconds on 4G LTE, both within the sub-3-second target. Latency on throttled 3G averaged 4.1 seconds; in fallback rule-only mode (no AI call required), this reduced to 1.2 seconds on all network types. Twilio SMS fallback delivery averaged 6.3 seconds end-to-end across all network conditions.

C. Detection Accuracy

Table 5 summarizes detection accuracy across scenario categories. The overall accuracy rate was 89% (26.7/30 correct classifications, averaged across risk tiers). High-risk scenarios achieved 90% true-positive detection with zero false positives. Medium-risk scenarios detected 80% of true threats with two false positives caused by extended stationary periods in dimly lit but objectively safe residential areas. Low-risk scenarios produced one false positive attributed to GPS signal multipath in an urban canyon.

Table 5: Detection Accuracy by Risk Category

Category	TP Rate	FP Rate	Avg Score
Low-Risk (n=10)	0%	7% (1/10)	18.3 ± 6.2
Medium-Risk (n=10)	80% (8/10)	20% (2/10)	52.1 ± 11.4
High-Risk (n=10)	90% (9/10)	0%	78.6 ± 9.8
Overall	89%	9%	—

IJARCCE ISSN (O) 2278-1021, ISSN (P) 2319-5940

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

D. Battery Performance

Over an 8-hour background monitoring session with the screen off and a mix of idle, moving, and one simulated high-risk episode, average battery drain was 9.3% on the mid-range device, 8.1% on the upper-mid-range device, and 7.6% on the flagship device. During sustained high-risk monitoring at 1-second GPS polling intervals, drain increased to approximately 21% per hour on the mid-range device—above the practical threshold for extended use. This confirms that adaptive sampling is necessary for full-day deployability and that sustained high-risk mode should be designed as a short-duration state.

E. AUC and Model Comparison

The hybrid scoring engine achieved an AUC of 0.94 across the 30 test scenarios. Rule-only scoring achieved an AUC of 0.87; AI-only scoring achieved 0.91. The improvement of the hybrid model over AI-only scoring is attributable to the stability provided by the rule-based Phase 1 in scenarios where Gemini's probabilistic output exhibited higher variance. The improvement over rule-only scoring is attributable to Gemini's ability to capture contextual nuance—particularly time-of-day and location semantics—that weighted rules cannot encode with comparable precision.

F. Usability (Pilot Study, n=15)

A pilot study with 15 female participants aged 19–28 from urban and semi-urban areas of Lucknow yielded a System Usability Scale score of 82.4, classified as Excellent. Thirteen of fifteen participants reported increased perceived safety when walking alone at night after using the application for one week. The Fake Call module was rated the most immediately practical feature by 9 of 15 participants. Two participants expressed initial concern about background location monitoring, both of which were resolved after reviewing the data minimization policy during onboarding. All trusted contacts correctly interpreted push notifications and accessed the live tracking link without guidance.

Table 6: Evaluation Summary

Metric	Result	Target
Alert Latency (Wi-Fi)	1.8 s avg	< 3 s
Alert Latency (4G)	2.7 s avg	< 3 s
Detection Accuracy	89%	> 85%
False Positive Rate	9%	< 10%
Battery (8h background)	9.3% avg	< 15%
Offline Recovery	< 8 s	< 10 s
AUC — Hybrid Engine	0.94	> 0.90
SUS Usability Score	82.4 / 100	≥ 75

XI. DISCUSSION

A. Architectural Insights

The hybrid Danger Score Engine's consistent outperformance of both rule-only (AUC 0.87) and AI-only (AUC 0.91) baselines validates the core architectural decision to fuse deterministic and probabilistic methods. The rule-based Phase 1 contributes stability and speed—it always produces a result, in milliseconds, with no network dependency. The Gemini AI Phase 2 contributes contextual nuance: it can reason about the semantics of a location type, the social meaning of unusual stationary behavior at a particular hour, or the combined significance of multiple low-weight features that individually fall below rule thresholds. Neither layer alone captures what the hybrid produces together.

The edge-cloud partitioning also has practical advantages beyond accuracy. By running Phase 1 entirely on-device, the system provides meaningful protection even during complete cloud outage. The AI layer enhances but does not enable protection—a philosophically important design choice for safety-critical applications where partial failure must remain functional.

B. The Fake Call and Evidence Vault as UX Safety Design

The Fake Call and Evidence Vault modules represent a category of safety design that goes beyond sensor fusion and alert dispatch. They address the social and behavioral dynamics of threatening situations. A woman who can plausibly receive a call from her mother has a credible, face-saving exit from a situation that has become uncomfortable but has not yet

DOI: 10.17148/IJARCCE.2026.15549

escalated to violence. This exit mechanism does not require her to signal distress, involve bystanders, or take any action that might provoke escalation. The decoy screen during Evidence Vault recording operates on the same logic: the danger of being seen to record is real, and designing around that danger is as important as the technical capability of the recorder itself.

These features emerged from qualitative user interviews conducted during the design phase, in which participants consistently described the social friction of appearing to use a safety app in public as a barrier to engagement. Designing for this friction—rather than ignoring it—produces tools that users are actually willing to deploy in real conditions.

C. False Positive Management

The 9% overall false positive rate, while within the target threshold of 10%, requires active management in production to prevent trust erosion. SafeHer addresses this through three mechanisms. First, a user-cancel window (default: 10 seconds in the safety verification dialog, 30 seconds for autonomously triggered full alerts) provides a low-friction correction path before notifications are dispatched. Second, post-alert feedback collection—prompting users to mark events as 'False Alarm' or 'Confirmed Threat'—feeds a retraining pipeline that adjusts rule weights and Phase 1 thresholds over time. Third, personalized threshold calibration based on individual behavioral baselines reduces false positives for users whose normal patterns (late-night commutes, low-density residential areas) would otherwise trigger frequent alerts.

D. Privacy as a Design Constraint, Not a Feature

The system's privacy architecture—differential privacy for the Fear Map, 24-hour raw GPS deletion, access-controlled Firestore rules, and decoupled AI prompts that receive features rather than raw coordinates—was not added after the core functionality was designed. It was imposed as a constraint from the beginning, which meaningfully shaped the architecture. The decision to send feature vectors rather than raw trajectory data to the Gemini API, for instance, reduces the AI system's exposure to precise movement history while providing sufficient context for accurate risk assessment. This approach offers a replicable template for location-based services operating under India's Digital Personal Data Protection Act and equivalent frameworks.

XII. LIMITATIONS

Battery consumption during sustained high-risk monitoring (1-second GPS polling) increases to 20–25% per hour on mid-range devices, limiting the practical duration of high-sensitivity operation. Android's Doze mode and iOS's Background App Refresh restrictions may suspend background services on devices with aggressive power management profiles, requiring explicit foreground service permissions that some users decline during onboarding.

The AI scoring component introduces a cloud dependency that cannot be fully eliminated under the current architecture. In areas with chronically degraded connectivity—semi-urban peripheries, basement transit stations, rural transit corridors—Phase 2 may be unavailable precisely when the risk of encountering danger is highest. The rule-based fallback provides meaningful degraded-mode protection, but the precision gap between hybrid and rule-only scoring (AUC 0.94 vs. 0.87) represents a real reduction in protective effectiveness.

GPS accuracy degrades substantially in dense urban environments due to signal multipath, causing transient spikes in the isolation index that produce false positives. Accelerometer thresholds for motion anomaly detection were calibrated on three test devices and may require per-device tuning to maintain accuracy across the wide range of Android hardware in the Indian market. The pilot study involved 15 participants in a single geographic region; broader demographic, linguistic, and geographic validation is needed before generalization.

Large language models including Gemini are probabilistic and may produce inconsistent risk assessments for identical context vectors across API calls. The system mitigates this through score averaging over a rolling window, but deterministic scoring guarantees cannot be made. Cultural and regional variation in what constitutes a semantically 'high-risk' location or time may also affect Gemini's scoring accuracy in regions beyond Lucknow without prompt engineering adjustments.

XIII. FUTURE WORK

A. On-Device AI and Edge Inference

Deploying a quantized gradient-boosted decision tree or lightweight transformer model directly on the device using TensorFlow Lite would eliminate the cloud latency and availability dependency of Phase 2, reducing alert end-to-end latency to under 500 milliseconds and enabling full operation in offline and airplane-mode scenarios. Federated learning would aggregate model weight updates from across the user base without transmitting raw sensor data to a central server, enabling continuous accuracy improvement while preserving the privacy architecture. Mobile Neural Processing Units available in Snapdragon 8 and Dimensity 9000 series SoCs are already capable of running inference workloads of this complexity without measurable battery impact.

International Journal of Advanced Research in Computer and Communication Engineering
Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026
DOI: 10.17148/IJARCCE.2026.15549

B. Wearable and IoT Integration

Bluetooth Low Energy integration with Wear OS and watchOS devices would enable motion data offloading to the wearable, freeing the phone's processor and battery. Heart rate variability monitoring from a paired smartwatch provides a physiological stress signal that could substantially improve threat detection accuracy when combined with motion and location features—a user whose HRV indicates acute stress while simultaneously stationary in an isolated location presents a qualitatively different risk profile from one who is simply standing still. eSIM-enabled wearables could provide independent alert dispatch capability when the paired phone has been taken or powered off.

C. Institutional Integration

Partnership with India's 112 emergency dispatch infrastructure would enable forwarding of verified, high-confidence SafeHer alerts directly to nearest-available responders, along with structured context data and a live GPS tracking link. This requires formal data-sharing agreements, institutional audit partnerships, and cryptographic alert signing to prevent spoofed emergency requests. Anonymized Fear Map data aggregated at the city level could inform municipal decisions about street lighting placement, CCTV coverage gaps, and dynamic police patrol routing—extending the platform's value from individual protection to urban safety infrastructure.

D. Personalized Behavioral Modeling

Individual behavioral baselines—characteristic walking speeds, frequent routes, usual active hours, regularly visited locations—would allow the scoring engine to define deviation from personal normal rather than deviation from population averages. A user who regularly walks home at 11 PM through a quiet residential lane should not receive the same time-risk and isolation scores as a user for whom that route and hour represent genuine anomalies. Personalized thresholds would reduce false positives for users with unconventional routines while maintaining sensitivity for genuinely unusual situations.

XIV. CONCLUSION

SafeHer demonstrates that the reactive safety paradigm—press a button, send an alert—can be replaced with something fundamentally more reliable. By continuously monitoring contextual signals and combining deterministic rule-based edge scoring with Google Gemini AI's contextual reasoning, the system identifies and responds to escalating threats without requiring the user to take any action. This removes the single greatest failure mode of existing personal safety tools: the assumption that a person in danger can and will initiate her own rescue.

The platform's scope extends beyond automated alerting. The Fake Call module addresses the social dynamics of low-escalation threatening situations, giving users a credible exit that does not signal distress. The Evidence Vault's decoy screen design reflects the real danger of being seen to document an incident. The Fear Map and Trust Score system extend individual situational awareness into a community intelligence layer. Together, these features constitute a safety ecosystem rather than a single alert mechanism.

Empirical evaluation across 30 simulated scenarios demonstrates 89% detection accuracy, sub-3-second alert latency across all tested network conditions, 9.3% average battery consumption over eight hours, and a System Usability Scale score of 82.4. The hybrid scoring engine outperforms both rule-only and AI-only baselines, validating the architectural decision to combine both approaches rather than rely on either alone.

The path forward—on-device AI inference, wearable integration, institutional emergency system connectivity, and personalized behavioral modeling—will extend SafeHer's protective capabilities while further reducing its dependency on network connectivity and cloud infrastructure. What this work establishes is the foundation: a production-grade, empirically validated demonstration that mobile devices can serve as silent, intelligent guardians, transforming personal safety from a reactive afterthought into a continuous, context-aware protective layer.

ACKNOWLEDGMENT

The authors express sincere gratitude to Dr. Nikhat Akhtar for her mentorship and technical guidance throughout this research. We thank the Department of Computer Science and Engineering, Goel Institute of Technology and Management, Lucknow, for providing the infrastructure and academic environment that made this work possible. We also thank the fifteen pilot participants whose candid feedback shaped the final design of SafeHer.

REFERENCES

[1] WHO, Violence Against Women: Global Estimates 2018, WHO Press, 2018.

[2] S. K. Dhillon and R. K. Singh, "Analysis of women safety apps," Int. J. Comput. Sci. Inf. Technol., vol. 12, no. 4, pp. 112–118, 2021.

[3] A. B. M. S. Uddin et al., "Context-aware mobile safety systems," IEEE Access, vol. 10, pp. 45123–45139, 2022.

[4] Google LLC, Flutter Documentation, 2024. https://docs.flutter.dev

International Journal of Advanced Research in Computer and Communication Engineering

Impact Factor 8.471 ⁜ Peer-reviewed & Refereed journal ⁜ Vol. 15, Issue 5, May 2026

DOI: 10.17148/IJARCCE.2026.15549

[5] Firebase Team, Firebase Cloud Firestore & Auth Documentation, 2024. https://firebase.google.com/docs

[6] Google DeepMind, Google Gemini API Documentation, 2025. https://ai.google.dev/gemini-api/docs

[7] M. Chen et al., "Edge-cloud collaborative AI for mobile anomaly detection," IEEE Trans. Mobile Comput., vol. 23, no. 5, pp. 4102–4115, 2024.

[8] R. K. Gupta and P. Sharma, "Privacy-preserving location tracking in safety apps," Proc. ACM Security Workshop, pp. 67–74, 2023.

[9] J. Park et al., "Differential privacy for urban safety heatmaps," IEEE ICDM, pp. 312–320, 2023.

[10] National Institute of Justice, Technology in Violence Prevention, U.S. Dept. of Justice, 2022.

[11] K. S. Rao et al., "Hybrid rule-AI models for contextual risk assessment," Sensors, vol. 24, no. 8, Art. 2511, 2024.

[12] T. A. Nguyen and L. M. Tran, "Battery-optimized background sensing," ACM MobiCom Workshop, pp. 45–52, 2023.

[13] R. S. Patel and D. K. Singh, "Federated learning for mobile safety," IEEE Access, vol. 12, pp. 78901–78912, 2024.

[14] National Crime Records Bureau, Crime in India 2021: Statistics, Ministry of Home Affairs, Government of India, 2022.

[15] J. Wei et al., "Emergent abilities of large language models," arXiv:2206.07682, 2023.

[16] L. Zhang and W. Liu, "Geofencing limitations in dynamic urban environments," IEEE Pervasive Computing, vol. 22, no. 3, pp. 45–53, 2023.

[17] T. Kim et al., "WalkSafe: A pedestrian safety app using real-time mobile sensing," Proc. ACM MobiSys, pp. 215–228, 2022.

[18] European Data Protection Board, Guidelines on Location Data & Privacy in Mobile Applications, 2023.

[19] IEEE Std 1620-2023, Standard for Safety-Critical Mobile App Architectures, IEEE, 2023.

[20] Apple Inc., iOS Background Execution & Power Management Guidelines, 2024.

[21] R. K. Yadav et al., "SafeHer: Design and Implementation of a Proactive AI Safety Platform," GITM Technical Report, 2025.

[22] A. Upadhyay et al., "Fake Call and Evidence Vault: Novel UX Safety Mechanisms for Women's Protection Apps," Proc. IndiaHCI, 2025.

