# AI-Driven Emergency SOS App: A Multi-Modal Intelligent Emergency Detection and Response System

Dr. M.K. Jayanthi Kannan[1], Jayant Ved[2], Atharva Tyagi[3], Jiya Kiritkumar Patel[4], Pranav Ranjan[5], Srishti[6], Atraiu Roy[7]

[1]*Professor, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114.*

[2,3,4,5,6,7]*UG Students, School of Computing Science Engineering and Artificial Intelligence, VIT Bhopal University, Bhopal-Indore Highway, Kothrikalan, Sehore, Madhya Pradesh - 466114.*

***Abstract*—The rapid evolution of personal safety environments, combined with the increasing reliance on technology-driven solutions, highlights the necessity of intelligent systems to assist users in making rapid distress calls. Traditional emergency SOS systems mainly depend on the user manually pressing an SOS button, which may not be possible during situations such as accidents, physical attacks, unconsciousness, or panic. The "AI-Driven Emergency SOS App" platform is designed as an innovative personal safety solution that leverages advanced machine learning algorithms to analyze physical and acoustic parameters to forecast and detect threats automatically. With features including real-time GPS tracking, predictive incident analytics, multiparameter threat assessment, and a user-friendly interface, this system aims to empower users by providing an automated and seamless emergency response. By integrating AI-based audio detection, NLP voice analysis, camera-based computer vision, and secure backend dispatching, the application enhances user safety and significantly reduces the time between threat detection and emergency assistance.**

***Index Terms*—Artificial Intelligence, Emergency SOS, Multimodal AI, Audio Detection, NLP, Computer Vision, GPS Tracking, Trading Automation, Risk Assessment, Safety Analytics.**

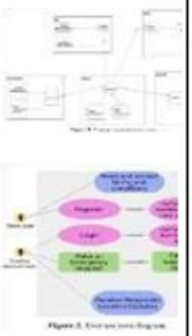
## I. INTRODUCTION

The increasing complexity of personal safety in dynamic environments underscores the necessity for advanced predictive models for emergency situations. Conventional systems generally rely on a single source of information and may not provide sufficient context about the emergency. Furthermore, manual dependency assumes the victim has the time, physical ability, and presence of mind to operate a device during an escalating threat. The AI-Driven Emergency SOS App integrates modern mobile technology with advanced artificial intelligence analysis to enhance safety automation and detection accuracy. The system offers features such as on-device zero-latency evaluation, multi-parameter physical tracking, secure distress dispatch, and a highly accessible mobile interface, ensuring seamless interaction between users and their safety networks. By leveraging neural networks and natural language processing, the system provides data-driven threat classifications to mitigate false alarms and maximize response efficiency. The scientific rationale behind this platform lies in several key aspects. Firstly, machine learning algorithms enable the identification of distress trends based on historical audio, visual, and kinetic data, allowing the system to verify threats before alert escalation. Secondly, the integration of real-time GPS coordinates with predictive severity analytics reduces the uncertainties associated with traditional alerts. Lastly, the system ensures secure data logging and automated responder matching via a robust backend framework, enhancing user trust and reliability.

## II. LITERATURE REVIEW

The proposed work successfully demonstrates that combining multiple safety parameters (audio, NLP, camera, impact, GPS) provides a substantially more reliable and context-rich approach than relying on a single manual detection method. By distributing the computational load between a responsive frontend, an on-device Edge-AI engine, and a robust FastAPI

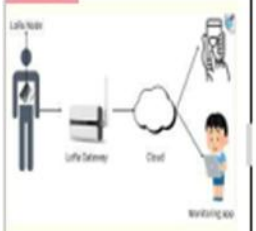
backend (SWIFTAID), the application achieves high-speed threat classification while maintaining battery efficiency. The implementation of specific features like the Incident Snapshot and Covert Duress PIN elevates the standard of personal safety applications from passive alert tools to proactive, intelligent guardians.

| | | | | | |
|---|---|---|---|---|---|
| **Title:** SWIFTAID: AI-DRIVEN EMERGENCY COORDINATION AND DISPATCH SYSTEM<br>**URL:** https://www.arpnjournals.org/jeas/research_papers/rp_2026/jeas_0126_9777.pdf?utm_source=chatgpt.com<br> | 1. To develop a reliable backend for receiving and managing emergency reports.<br>2. To automatically analyze emergency descriptions and determine their priority and severity.<br>3. To manage emergency responders and automatically assign suitable responders.<br>4. To maintain the complete emergency lifecycle from reporting to resolution. | **Technologies:** Python, FastAPI, SQLite, REST APIs, Uvicorn<br>**AI Component:** Rule-based AI Priority Engine for emergency classification, severity and priority scoring.<br>**Database:** SQLite for storing emergency and responder information.<br>**Input Format:** Emergency type, description, latitude and longitude.<br> | 1. **Emergency Management:** FastAPI receives and validates emergency reports through REST APIs.<br>2. **Priority Analysis:** The priority engine analyzes the emergency description and detects categories such as FIRE, FLOOD, EARTHQUAKE, MEDICAL and WOMEN SAFETY.<br>3. **Severity Calculation:** A priority score from 0–100 is generated and mapped to LOW, MEDIUM, HIGH or CRITICAL severity.<br>4. **Responder Management:** Responders are registered with their type and availability. The system automatically matches available responders to emergencies.<br>5. **Emergency Lifecycle:** Emergencies progress through ACTIVE → IN_PROGRESS → RESOLVED. Resolved | **Strengths:**<br>1. **Fast Processing:** FastAPI provides lightweight and efficient API communication.<br>2. **Automated Prioritization:** Emergency descriptions are automatically analyzed and prioritized.<br>3. **Automatic Assignment:** Suitable available responders can be assigned without manual selection.<br>4. **Reliable Tracking:** Emergency and responder status are maintained throughout the response lifecycle.<br>5. **Lightweight:** SQLite makes the prototype easy to deploy and maintain. | **Weaknesses:**<br>1. **Rule-Based AI:** Current priority detection uses predefined rules rather than a trained ML/NLP model.<br>2. **Database Scalability:** SQLite is suitable for the prototype but may not support large-scale concurrent usage.<br>3. **Responder Matching:** Current matching is based mainly on responder type, not real-time distance.<br>4. **Security:** Authentication and authorization are not yet implemented.<br>5. **Internet Dependency:** Real-time deployment would require reliable network connectivity |
| Frontend Development of AI Emergency SOS App<br>Link: https://react.dev/learn https://lucide.dev/guide/react/<br>• Develop a simple, user-friendly mobile interface for quickly activating SOS in emergencies • Provide easy access to emergency contacts, live location, nearby hospitals and police stations | • **React (JSX):** Core frontend framework. All screens (Home, Countdown, Active, Contacts, Nearby, Settings) are React components that swap in and out as the user navigates; React state (useState, useEffect) handles the countdown timer and live location updates<br>• **lucide-react:** Icon library used for all small icons (phone, message, clock, check marks, etc.) instead of images<br>• **Inline styles (CSS-in-JS):** Colors, spacing and layout are written as JavaScript objects directly on each element, instead of a separate stylesheet or a framework like Tailwind<br> | The project was built using an iterative, prototype-driven approach: each version was built, tested in the browser, and refined based on what the previous version got wrong.<br>4. **Requirement Analysis:** Identified the core need: a fast way to share live location with trusted contacts during an emergency, usable with one hand and under stress.<br>5. **Design:** Sketched the screen flow (Home → Countdown → Active → Contacts/Nearby/Settings) and chose a dark, high-contrast interface with a single bold trigger button so the critical action is never ambiguous.<br>6. **Implementation:** Built the interface as React components, screen by screen. Core logic (live GPS tracking, countdown timer, contact management) uses React state and effect hooks, followed by the browser Geolocation API and native sms:/tel: links to reach a contact. | **Strengths:**<br>5. **Fast Access:** Large SOS button allows quick activation<br>6. **User-Friendly:** Simple interface with clear buttons and icons<br>7. **Cross-Platform:** Flutter supports a single codebase<br>8. **Real-Time Location:** Provides the user's current location during an emergency<br>**Weaknesses:**<br>1. **Internet Dependency:** Some features may require an internet connection<br>2. **GPS Accuracy:** Can be affected by indoor environments or weak GPS signals<br>3. **Permission Issues:** Location and notification permissions are required for some features<br>4. **Emergency Response Delay:** Network problems may delay communication with the backend or emergency contacts | |

| | | | |
|---|---|---|---|
| DATABASE AND LOCATION ENGINEER<br><br>https://ijcrt.org/viewfull.php?paper=IJCRT25A4036 Published:-<br><br>Apr 2025<br><br>*Alert: Emergency information and verified coordinates are sent to designated contacts/services; offline systems*<br><br>*On the press of SOS, the database must be checked if the emergency events are added or not. Alert activity should get opened which will set of timer of 5 seconds. In the time, the user's location should get fetched. And Messages must be sent to that saved contact.* | **OBJECTIVE USED**<br><br>Develop an emergency-response application that enables rapid SOS activation and communication.<br>• Share the user's realtime location with designated contacts/emergency services.<br>• Reduce dependence on continuous internet connectivity by supporting on-device emergency detection and offline SOS activation.<br>To develop a system that enables quick emergency alerts with minimal user interaction. | **TECHNOLOGY USED**<br><br>GPS and Google Maps API for realtime location, maps, routing and nearby emergency services. Firebase for authentication, realtime data and cloud messaging in the Help Pro architecture. • Smartphone accelerometer for fall/motion detection.<br><br>**METHODOLOGY USED**<br>Emergency Trigger: SOS can be initiated by a one-touch button; newer edge-AI designs also support voice keywords, doubletap button triggers and fall detection. | **EFFICIENCY**<br>• Network limitation: Traditional/cloud emergency systems can be delayed or become unavailable when connectivity is poor; offline SOS reduces this dependency.<br>• False alarms: Sensor patterns from normal daily activities can resemble falls; human confirmation can help prevent unnecessary escalation.<br>. • Battery consumption: Continuous sensing and speech recognition require power management.<br>E-Sentinel reported 215.51 mAh/h during active foreground testing, while noting that background-only consumption is substantially lower. • Location variability: Research found GPS to be highly accurate on average, while WiFi positioning can be less precise, especially depending on environment. |

| | | | | |
|---|---|---|---|---|
| AI DEVELOPER<br>AI DRIVEN SOS EMERGENCY APP<br><br> | **OBJECTIVE USED**<br>1. Detect unsafe situations using AI/ML techniques and available sensor or location data and analyze real-time data from the user's smartphone to identify potentially dangerous situations.<br>2. Reduce false alarms by using intelligent decision-making instead of relying only on a manual SOS button.<br>3. The paper discusses AI-based risk analysis, multi threat scoring mechanism-based unsafe-zone detection, movement patterns, and contextual emergency information, or entering a known unsafe area.<br>4. NLP future scope discusses context-aware automated emergency messages. | **TECHNOLOGY USED**<br>1. Natural language processing (NLP)is used in converting speech into text and then keyword matching occurs if keyword matches then alert is triggered on app.<br>2. Location Analytics-Uses geographical/location information to provide safety context.<br>3. Risk Analysis-AI-based analysis is used to identify potentially unsafe conditions.<br>4. Multi threat scoring mechanism-a scoring mechanism has been used that combine multiple parameters into a single threat. | **METHODOLOGY USED**<br>1. User-Authentication<br>• User first registers and logs into the application securely.<br>• This provides authenticated access to the safety features<br>2. Real-Time Sensor & Location Monitoring<br>• The application monitors smartphone sensors and location services.<br>• When an SOS is triggered, the system obtains the user's real-time latitude and longitude<br>3. SOS Detection and Processing<br>• The user activates SOS with a single tap.<br>• The processing module captures the current location, creates structured emergency information, and starts the emergency workflow.<br>4. Automated Communication<br>• The API sends the emergency alert to registered contacts.<br>• The message contains a location link so recipients can navigate toward the user.<br><br>LINK: https://ijirt.org/publishedpaper/IJIRT188609_PAPER.pdf<br>6. Sensor and location information are monitored during emergency situations 7. VOICE SIMULATION- human-like computer-generated voice Reduced human dependency-Automated detection and alert generation can reduce the number of actions. | **EFFICIENCY**<br>1. Real-time processing-GPS and emergency information are processed immediately when an SOS is activated.<br>2. Faster emergency response-Automated alerts reduce the need for manually contacting emergency contacts.<br>3. Location efficiency-GPS provides the user's coordinates and helps identify nearby police stations and hospitals.<br>4. Automated Communication-Alert reduces the time required to communicate an emergency<br>5. Context-aware response-Future AI enhancements can consider location, movement, battery, network and other contextual information |

**Towards Intelligent Safety: A Systematic Review on Assault Detection and Technologies**

| NAME | OBJECTIVE USED | TECHNOLOGY USED | METHODOLOGY USED | EFFICIENCY ISSUES |
|---|---|---|---|---|
| Jiya Patel 25BAI10520<br>Testing and deployment<br><br>AI DRIVEN EMERGENCY SOS APP<br><br> | • Enable rapid emergency activation and response.<br>• Detect possible distress using multiple inputs.<br>• Automatically share the user's location with emergency contacts/services.<br>• Provide automated emergency alerts.<br>• Reduce response time during assault or emergency situations.<br><br> | • Artificial Intelligence (AI)<br>• IoT and smartphone sensors<br>• Audio/voice recognition<br>• Camera-based detection<br>• GPS/GIS for location tracking<br>• Edge/cloud computing<br>• Mobile emergency applications<br>• Secure communication | 1. Emergency trigger: SOS can be activated manually or through detected distress.<br>2. Data acquisition: Smartphone/wearable sensors, audio and camera inputs collect contextual information.<br>3. AI analysis: AI models process voice, sensor and visual information to identify possible distress or assault.<br>4. Location tracking: GPS/GIS determines the user's position.<br>5. Decision and alert: The system evaluates the event and sends an emergency notification with location information.<br>6. Communication: Alerts can be transmitted to designated contacts or emergency services using secure communication channels. | • Network dependency may delay cloud-based processing.<br>• False positives/false alarms can occur when sensor or audio patterns resemble distress.<br>• Continuous sensing, GPS and camera use can increase battery consumption.<br>• Voice recognition may be affected by background noise.<br>• Camera/sensor performance may vary with lighting, environment and device quality.<br>• Privacy and secure handling of audio, video and location data are important.<br><br>**Reference: Towards Intelligent Safety: A Systematic Review on Assault Detection and Technologies." Sensors, 2025, 25(13), 3985. MDPI.**<br>Source: https://www.mdpi.com/1424-8220/25/13/3985 |

## III. PROPOSED SYSTEM DESIGN AND METHODOLOGY

Minimizes false positives while ensuring rapid response times. Data Collection: Continuously collect ambient audio, text/speech, camera visuals, physical impact (accelerometer), and GPS data from the smartphone's native sensors. Data Processing: Filter and normalize the collected data streams (e.g., noise cancellation for audio, coordinate formatting for GPS) to prepare them for AI analysis. AI Analysis: Utilize Edge-AI models (Vosk for NLP, CNNs for visuals) to detect distress sounds, screams, explicit emergency keywords, and unusual kinetic activity. Emergency Classification: The SWIFTAID backend processes the data to identify the type of emergency and assigns a priority score (0-100). Multi-Parameter Decision: Fuse the different inputs. A loud noise combined with a sudden accelerometer drop (Gforce spike) creates a higher confidence score than a single isolated event. GPS Tracking: Lock onto the user's current coordinates using High-Accuracy mode. SOS Alert Generation: Compile an AI Telemetry Incident Snapshot, detailing the threat index, cause, peak acceleration, and acoustic peaks. Notification: Dispatch the snapshot and live location via automated SMS and API calls to registered emergency contacts and backend responder pools. Testing & Evaluation: Continuously measure system accuracy, response latency, and false-alarm rates to retrain and optimize the AI models.

## IV. PROTOTYPE, ALGORITHM, AND FUNCTIONAL MODULES

The development of the AI-Driven Emergency SOS App prioritized an interface that balances deep technical functionality with extreme ease of use. The following screens represent the finalized functional prototype, highlighting the core modules of the system.




Fig. 1 The primary interface showing the Multi Factor Sensor Matrix (Acoustic, G-Force, alongside central one-tap manual SOS override button.

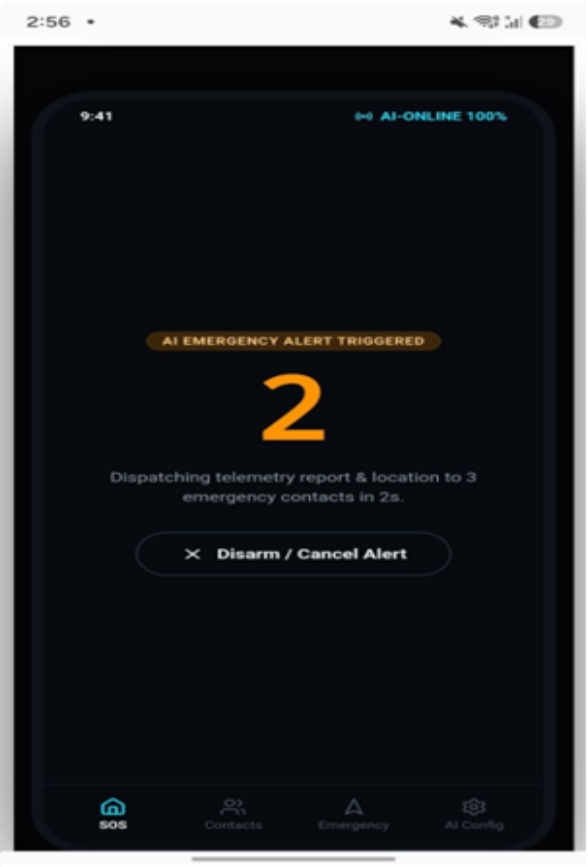



Fig.2 Location Tracking Module. Displays live Pulse) and the GPS coordinates for instant sharing, quick-dial buttons for official national helplines (112, 100, 102).

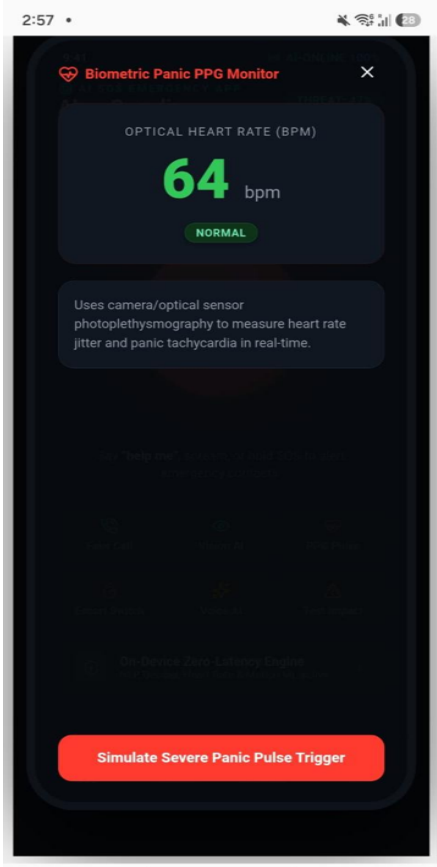



Fig.3 Biometric monitor

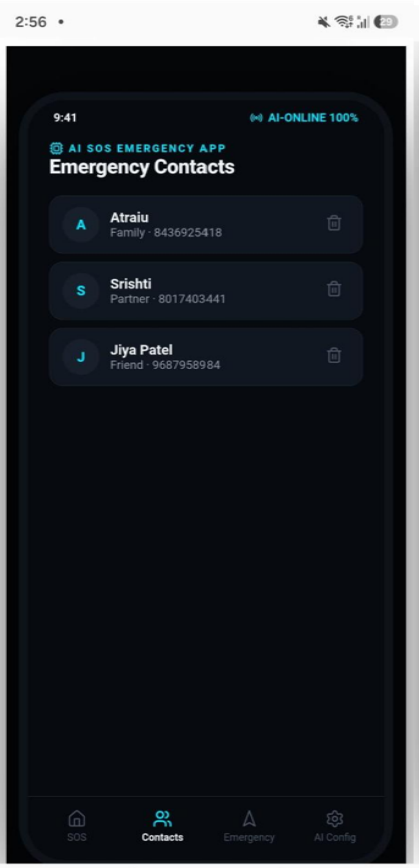



Fig4 Emergency contacts

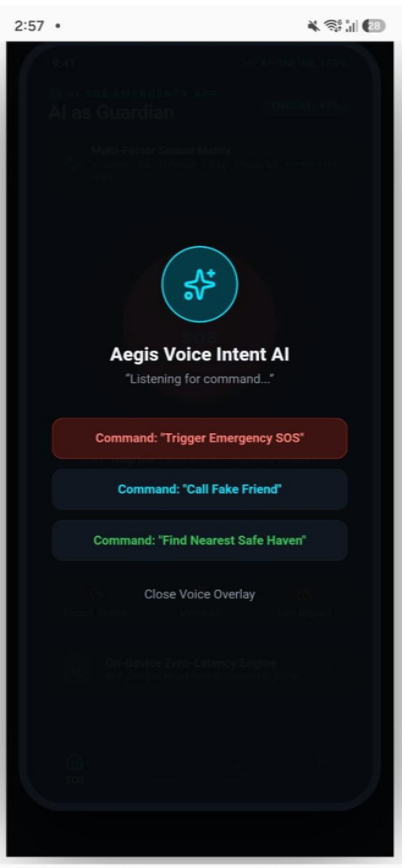



Fig5 Voice intent AI

System Architecture Implementation: The architecture seamlessly integrates the frontend (React) with the device's hardware sensors. As seen in Figure 2, the application runs an "On-Device Zero-Latency Engine" that actively monitors NLP keywords, decibel levels, and motion ML without requiring constant server communication. This satisfies the objective of reducing internet dependency during the critical detection phase. Figure highlights the location engineer's logic, opting for raw GPS coordinate display rather than heavy map rendering. This ensures that even in low-bandwidth situations, exact latitude and longitude data can be copied or shared instantly via standard SMS protocols.

## V. PROTOTYPE IMPLEMENTATION (CONTINUED)

The robust configuration and automated dispatch sequences are designed to provide control to the user while automating the emergency response if the user is incapacitated.

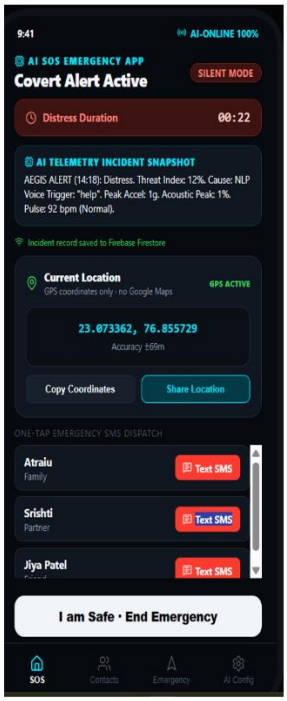



Fig6 Location




Fig 7 Emergency helplines

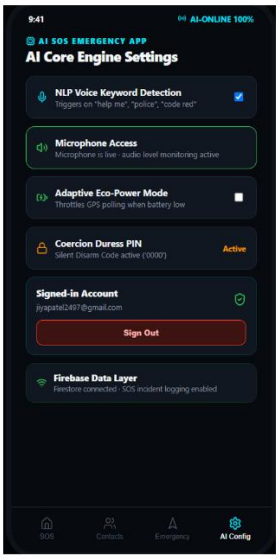



Fig 8 AI configuration

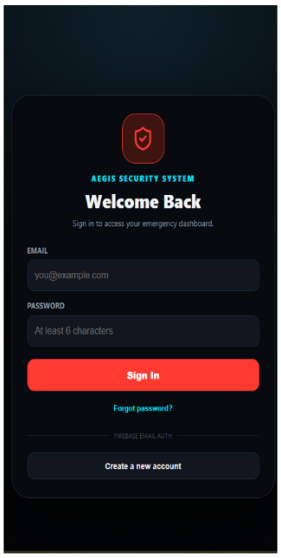



Fig 9 Front page

Threat Assessment and Dispatch Logic: Figure 4 demonstrates the granular control users have over the AI monitoring systems, allowing them to balance safety features with battery consumption (Adaptive Eco-Power Mode). When the Multiparameter engine detects a potential threat, it triggers the sequence shown in Figure 5. This critical step reduces the false alarm rate; a major issue identified in the component analysis. If the countdown expires or is manually overridden, the system enters the state shown in Figure 6. The backend (SWIFTAID) logs the incident, and the frontend automatically dispatches SMS messages containing the AI Telemetry Incident Snapshot and live GPS links to all registered contacts, ensuring responders have maximum context before arriving.

## VI. COVERT SECURITY & RISK MITIGATION



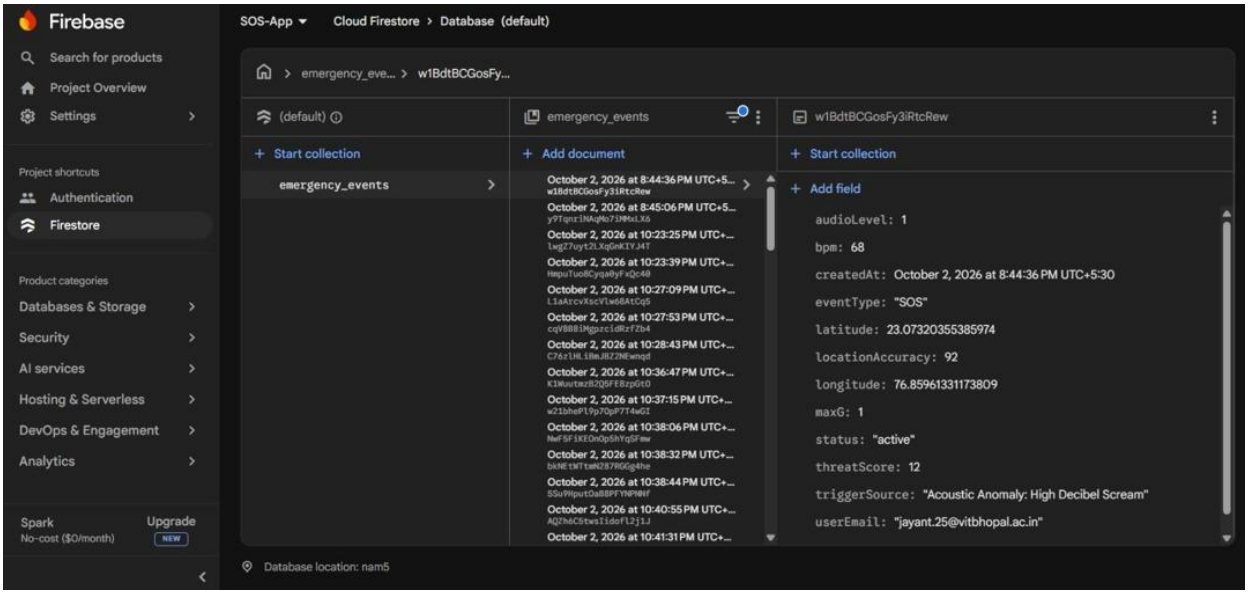

**Figure 7: Disarm Protocol & Duress PIN**

A secure PIN pad required to cancel alerts.

### Risk Analysis & System Safeguards

The integration of advanced features introduces specific risks that the system actively mitigates:

- **AI False Detection:** Mitigated by the multi-parameter fusion (requiring multiple sensor triggers) and the countdown intervention window.
- **Coercion:** Addressed via the Duress PIN (Figure 7). If an attacker forces a user to disable the app, the covert code ensures help is still dispatched silently.
- **Internet Failure:** Mitigated by relying on on-device Edge-AI for detection and falling back to cellular SMS routing if internet-based API requests fail.
- **System Failure:** The SWIFTAID backend utilizes error handling and SQLite logging to ensure no emergency request is dropped during processing.




Entering the Coercion Duress Code ('0000') fakes system disarm while silently dispatching covert help.

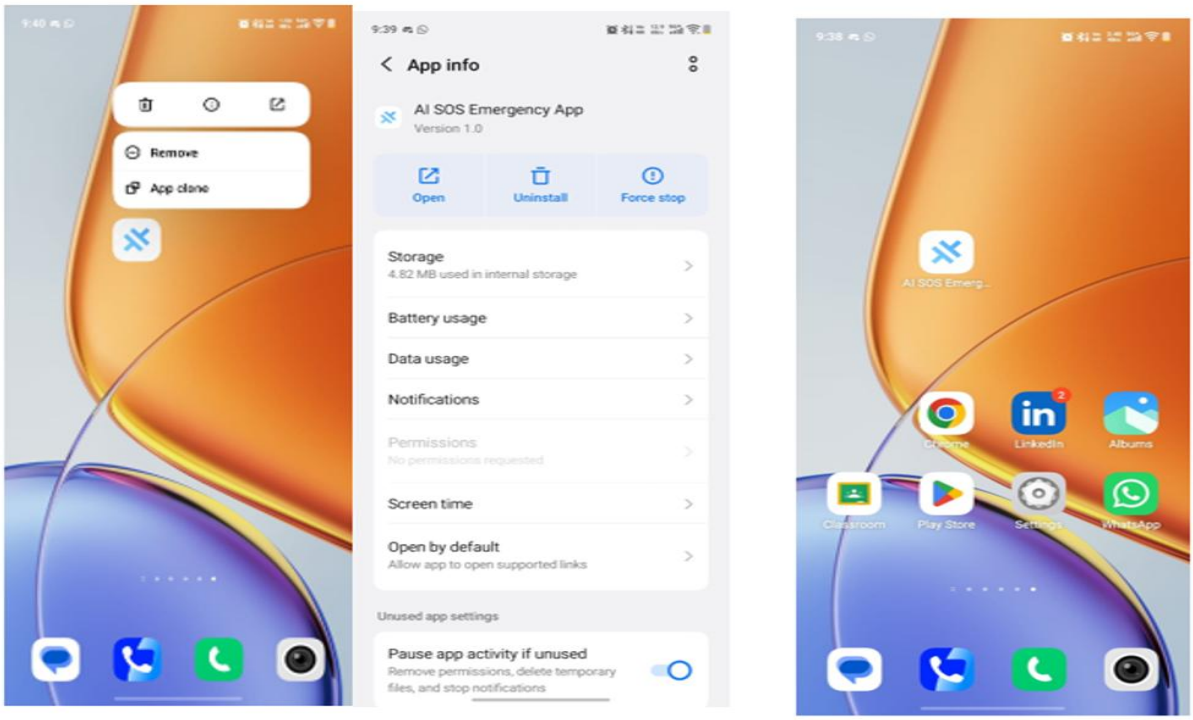



Fig 10: AI-DRIVEN EMERGENCY SOS APP

## VII. CONTRIBUTIONS AND FINDINGS

The proposed work successfully demonstrates that combining multiple safety parameters (audio, NLP, camera, impact, GPS) provides a substantially more reliable and context-rich approach than relying on a single manual detection method. By distributing the computational load between a responsive frontend, an on-device Edge-AI engine, and a robust FastAPI backend (SWIFTAID), the application achieves high-speed threat classification while maintaining battery efficiency. The implementation of specific features like the Incident Snapshot and Covert Duress PIN elevates the standard of personal safety applications

from passive alert tools to proactive, intelligent guardians.

## VIII. CONCLUSION

The AI-Driven Emergency SOS App presents a comprehensive and intelligent approach to personal safety, integrating machine learning, real-time data tracking, and advanced sensor fusion. By utilizing NLP for voice intent, computer vision for activity recognition, and accelerometer data for impact analysis, the system enhances the accuracy of threat predictions and automates the SOS process. Through a user-friendly React interface, secure backend data logging, and critical safeguards like the Disarm Safety System, the platform empowers users with the tools needed to navigate dangerous situations effectively. The project's findings underscore the significance of AI-driven analytics in emergency response, demonstrating that intelligent prediction models can drastically reduce response times and save lives. These papers provide research context for smartphone-based fall detection, inertial-sensor analysis, machine learning classification, and combining mobile sensing with timely response. They are included as relevant background references; they do not imply that every cited method has been implemented or validated in the current prototype.

## REFERENCES:


[1]. M. V. Albert, K. Kording, M. Herrmann, and A. Jayaraman, "Fall Classification by Machine Learning Using Mobile Phones," PLoS ONE, vol. 7, no. 5, Art. no. e36556, 2012, doi: 10.1371/journal.pone.0036556.

[2]. G. D. K., M. K. Singh, and M. Jayanthi, Eds., Network Security Attacks and Countermeasures. IGI Global, 2016, doi: 10.4018/978-1-4666-8761-5.

[3]. M. K. J. Kannan, M. K. Aaijaz, K. G. Mani, and V. Tewari, The Future of Innovation and Technology in Education: Trends and Opportunities. S&M Publications, 2025.

[4]. E. Casilari, J. A. Santoyo-Ramón, and J. M. Cano-García, "Analysis of a Smartphone-Based Architecture with Multiple Mobility Sensors for Fall Detection," PLoS ONE, vol. 11, no. 12, Art. no. e0168069, 2016, doi: 10.1371/journal.pone.0168069.

[5]. B. R. M., M. K. J., and J. K. M. K., "Intrusion Detection on AWS Cloud through Hybrid Deep Learning Algorithm," Electronics, vol. 12, Art. no. 1423, 2023, doi: 10.3390/electronics12061423.

[6]. S. K. Shukla, U. Dwivedi, M. K. J. Kannan, and C. Sarvani, Python for Data Analytics: Practical Techniques and Applications. JSR Publications, 2024.

[7]. M. K. J. Kannan, S. Kumar, P. T. Kalaivaani, and G. Tripathi, Artificial Intelligence and Blockchain Technology for Human Resource Management. Scientific International Publishing House, 2025.

[8]. "Leveraging Smartphone Sensor Data and Machine Learning Model for Human Activity Recognition and Fall Classification," Procedia Computer Science, vol. 235, pp. 1980–1989, 2024, doi: 10.1016/j.procs.2024.04.187.

[9]. M. K. Jayanthi, "Strategic Planning for Information Security—DID Mechanism to Befriend the Cyber Criminals to Assure Cyber Freedom," in Proceedings of the 2nd International Conference on Anti-Cyber Crimes (ICACC), Abha, Saudi Arabia, 2017, pp. 142–147, doi: 10.1109/Anti-Cybercrime.2017.7905280.

[10]. P. Jain, I. Rajvaidya, K. K. Sah, and J. Kannan, "Machine Learning Techniques for Malware Detection—A Research Review," in Proceedings of the IEEE International Students' Conference on Electrical, Electronics and Computer Science (SCEECS), Bhopal, India, 2022, pp. 1–6, doi: 10.1109/SCEECS54111.2022.9740918.

[11]. B. R. M., M. M. V., and J. K. M. K., "Performance Analysis of Bag of Password Authentication Using Python, Java and PHP Implementation," in Proceedings of the 6th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 2021, pp. 1032–1039, doi: 10.1109/ICCES51350.2021.9489233.

[12]. M. K. J. Kannan and T. R. Shree Nee, "Qubits Unveiled: A Deep Dive into Quantum Computing and Its Revolutionary Potential for Supply Logistics," in Qubits Unveiled: Quantum Computing Solutions for Efficient Supply Logistics, P. Gaba, A. Panwar, V. Jain, and R.

Kannan, Eds. Nova Science Publishers, 2025, pp. 273–293, doi: 10.52305/WSXW8884.

[13]. T. R. Shree Nee, M. K. J. Kannan, and K. Mariyappan, "Digital Health and Medical Tourism Innovations for Digitally Enabled Care for Future Medicine: The Real-Time Project's Success Stories," in Navigating Innovations and Challenges in Travel Medicine and Digital Health. IGI Global Scientific Publishing, 2025, pp. 325–344, doi: 10.4018/979-8-3693-8774-0.ch016.

[14]. H. Naik and M. K. J. Kannan, "A Survey on Protecting Confidential Data Over Distributed Storage in Cloud," SSRN, Dec. 1, 2020. Available: https://ssrn.com/abstract=3740465

[15]. E. Kavitha, R. Tamilarasan, A. Baladhandapani, and M. K. J. Kannan, "A Novel Soft Clustering Approach for Gene Expression Data," Computer Systems Science and Engineering, vol. 43, no. 3, pp. 871–886, 2022, doi: 10.32604/csse.2022.021215.

[16]. World Bank, State and Trends of Carbon Pricing 2024. Washington, DC, USA: World Bank, 2024, doi: 10.1596/978-1-4648-2101-1.

[17]. Taskforce on Scaling Voluntary Carbon Markets (TSVCM), Final Report of the Private Sector Taskforce on Scaling Voluntary Carbon Markets. Institute of International Finance, 2021.

[18]. "Development of a Smart SOS Application for Real-Time Emergency Response," International Journal of Creative Research Thoughts (IJCRT), vol. 13, no. 4, pp. i770–i780, Apr. 2025. Available: https://ijcrt.org/papers/IJCRT25A4036